

AURELIAN RĂDOUCU

*Probleme de informatică
pentru liceu*

MATEMATICĂ-INFORMATICĂ INTENSIV

CLASA A X-A

VARIANTA C++

PREFAȚĂ

Lucrarea urmărește noua programă școlară elaborată pentru clasa a XI-a, de la profilul matematică-intensiv informatică. Problemele prezentate în fiecare capitol vin în sprijinul cunoașterii și aprofundării conceptelor de programare.

Culegerea de față a fost concepută ca un auxiliar metodico-științific necesar elevilor în vederea realizării unei pregătiri temeinice pentru susținerea examenului de bacalaureat.

Lucrarea este structurată în cinci capitole, fiecare capitol conținând atât probleme rezolvate, probleme propuse , cât și subiecte model grile tip variantelor de bacalaureat . Prin problemele rezolvate am urmărit să formăm atât deprinderea de rezolvare corectă a algoritmilor, cât și abordarea diferențiată a acestora.

Culegerea se adresează de asemenea profesorilor de informatică, prin structură și conținut, fiind un instrument auxiliar de informare științifică pentru lecțiile curente, lecțiile recapitulative sau pentru pregătirea concursurilor școlare.

În plus, ceea ce recomandă culegerea de față este faptul că prezintă o paletă largă de probleme, organizate gradual, de la probleme obișnuite, uzuale la probleme complexe, originale.

Autor

CAPITOLUL I ȘIRURI DE CARACTERE

Itemii următori sunt preluați din variantele de bacalaureat 2009.

1. În secvența de instrucțiuni de mai jos, variabila `s` memorează un șir de caractere format doar din litere ale alfabetului englez, iar variabilele `i` și `n` sunt de tip `int`. Știind că în urma executării secvenței s-a afișat succesiunea de caractere `eied*ael*` scrieți care este șirul de caractere memorat de variabila `s`.

```
n=strlen(s);
for(i=0;i<n;i++)
if (s[i]=='e') cout<<'*';
else cout<<'e'<<s[i];
```

2. Fie `s` o variabilă ce memorează un șir de caractere, format doar din litere ale alfabetului englez, și `i` o variabilă de tip `int`. Scrieți instrucțiunile ce pot înlocui punctele de suspensie din secvența de program alăturată astfel încât executarea ei să determine eliminarea tuturor literelor mici din șirul `s` și apoi afișarea șirului obținut.

```
i=0;
while (i<strlen(s))
.....
cout<<s;
```

3. Fie `s` o variabilă ce memorează un șir de caractere, `c` o variabilă de tip `char`, iar `i` și `j` două variabile de tip `int`. Scrieți instrucțiunile ce pot înlocui punctele de suspensie din secvența de program alăturată astfel încât executarea ei să determine modificarea conținutului șirului `s` prin interschimbarea caracterelor aflate pe poziții simetrice față de mijlocul șirului (primului caracter cu ultimul, al doilea cu penultimul, etc).

```
i=0;
j=strlen(s)-1;
while (i<j)
{...}
```

4. Ce se afișează în urma executării secvenței de program alăturate dacă variabila `s` memorează șirul de caractere `abcdefgh`?

```
strcpy(s+2, s+4);  
cout<<s<<" "<<strlen(s);
```

5. Se consideră declararea de mai jos:

```
char s[50], x[50];
```

Ce se afișează în urma executării secvenței de program scrisă alăturat dacă variabila s memorează șirul abcdefg?

```
strcpy(x, s+4);  
strcpy(s+4, "123");  
strcat(s, x);  
cout<<s;
```

6. Ce se va afișa în urma executării secvenței de instrucțiuni alăturate dacă variabila s memorează șirul de caractere abbacdde, iar variabila i este de tip întreg?

```
i=0;  
while (i<strlen(s)-1)  
if (s[i]==s[i+1])  
strcpy(s+i, s+i+1);  
else  
i=i+1;  
cout<<s;
```

7. Ce se afișează pe ecran în urma executării secvenței de program alăturate, în care variabila s memorează un șir cu cel mult 12 caractere, iar variabila i este de tip întreg?

```
strcpy(s, "abracadabra");
```

```
i=0;  
cout<<strlen(s);  
while (i<strlen(s))  
if (s[i]=='a')  
strcpy(s+i, s+i+1);  
else  
i=i+1;  
cout<<" "<<s;
```

8. Scrieți ce se afișează pe ecran în urma executării secvenței de program alăturate, în care variabila s memorează un șir de cel mult 12 caractere, iar variabila i este de tip întreg.

```
char s[13]="informatica";
cout<<strlen(s);
for (i=0;i<strlen(s);i++)
if (strchr("aeiou",s[i])!=NULL)
s[i]= '*';
cout<<" "<<s;
```

9. Scrieți ce se afișează pe ecran în urma executării secvenței de program alăturate, în care variabila s memorează un șir de cel mult 12 caractere, iar variabila i este de tip întreg.

```
char s[13]="abcdefghoid";
i=0;
cout<<strlen(s);
while (i<strlen(s))
if (strchr("aeiou",s[i])!=NULL)
strcpy(s+i,s+i+1);
else i++;
cout<<" "<<s;
```

10. Ce se afișează pe ecran în urma executării secvenței de program alăturate, în care variabila s memorează un șir cu cel mult 10 caractere, iar variabilele i și j sunt de tip întreg?

```
char s[11]="abcduecda";
cout<<strlen(s);
i=0; j=strlen(s)-1;
while (i<j)
if (s[i]==s[j])
{ strcpy(s+j,s+j+1);
strcpy(s+i,s+i+1); j=j-2;
}
else
{ i=i+1; j=j-1; }
cout<<" "<<s;
```

11. Ce se va afișa pe ecran în urma executării secvenței de program alăturate, în care variabila s memorează un șir cu cel mult 10 caractere, iar variabila i este de tip întreg?

```
i=0; char s[11]="abaemeiut";
cout<<strlen(s);
while (i<strlen(s))
if (strchr("aeiou",s[i])!=NULL)
{ strcpy(s+i,s+i+1); i=i+1; }
else
i=i+2;
cout<<" "<<s;
```

12. Ce se va afișa în urma executării secvenței de program alăturate știind că variabila a memorează un șir cu cel mult 100 de caractere, iar variabilele i și k sunt de tip întreg?

```
k='a'-'A';
strcpy(a,"clasa a-XII-a A");
cout<<a<<endl;
for(i=0;i<strlen(a);i++)
if(a[i]>='a' && a[i]<='z') a[i]=a[i]-k;
cout<<a;
```

13. Ce se va afișa în urma executării secvenței de program alăturate știind că variabila a memorează un șir cu cel mult 100 de caractere, iar variabila i este de tip întreg ?

```
strcpy(a,"bacalaureat");
cout<<strlen(a)<<endl;
for(i=0;i<strlen(a);i++)
if(strchr("aeiou",a[i])!=0)
cout<<'*';
```

14. În secvența de program alăturată, variabila a memorează un șir cu cel mult 100 de caractere, iar variabila i este de tip întreg. Completați punctele de suspensie din secvență astfel încât, în urma executării secvenței, aceasta să afișeze șirul de caractere *nf*rm*t*c*.

```
strcpy(a,"informatica");
for(i=0;i<strlen(a);i++)
if(...)
```

```
cout<<...;
else
cout<<...;
```

15. Ce se va afișa în urma executării secvenței de program alăturate, știind că variabila x memorează un șir cu cel mult 100 de caractere, iar variabila i este de tip întreg ?

```
strcpy(x, "bac2009");
cout<<x<<endl;
for(i=0; i<strlen(x); i++)
if (strchr("0123456789", x[i])==0)
cout<<x[i];
```

16. În secvența de program alăturată, variabila a memorează un șir cu cel mult 100 de caractere, iar variabila i este de tip întreg. Completați punctele de suspensie, astfel încât, în urma executării secvenței, să se afișeze doar literele mici și literele mari din șirul de caractere memorat în variabila a.

```
strcpy(a, "Bac 2009 iulie");
for(i=0; i<strlen(a); i++)
if(...)
cout<<a
```

17. Ce valoare are expresia de mai jos dacă variabila s memorează șirul de caractere alfabet?

```
strlen(strcpy(s, s+2))
```

18. Care din următoarele expresii are valoarea 1 dacă și numai dacă șirul de caractere s, format din exact 10 caractere, este obținut prin concatenarea a două șiruri identice?

- a. strcmp(s, s+5)==0
- b. s==strstr(s, s+5)
- c. s==s+5
- d. strcmp(s, strcat(s, s+5))==0

19. Scrieți o secvență de instrucțiuni C/C++ care determină, în urma executării ei, afișarea pe ecran a mesajului Corect dacă un șir de maximum 100 caractere, memorat de variabila s, este palindrom sau

mesajul Incorect în caz contrar. Un șir de caractere este palindrom dacă citit de la început la sfârșit este identic cu șirul citit de la sfârșit la început.

Exemplu: șirul de caractere cojoc este palindrom

20. Care va fi șirul de caractere afișat după executarea secvenței de program alăturate, în care variabila s memorează un șir cu cel mult 5 caractere ?

```
char s[]="raton";
s[1]=s[3];
cout<<s;
```

21. Care va fi șirul de caractere afișat pe ecran după executarea secvenței de program alăturate în care variabila s memorează un șir cu cel mult 4 caractere iar variabila t un caracter?

```
char s[]="arac";
t=s[1]; s[1]=s[3];
s[3]='t';
cout<<s;
```

22. Care este valoarea expresiei strlen(s) pentru variabila s de tip șir de caractere, declarată și inițializată astfel:

```
char s[15]="Proba_E";
```

23. Ce se va afișa în urma executării secvenței alăturate, în care variabila c memorează un șir cu cel mult 20 de caractere, iar i este o variabilă de tip întreg?

```
char c[21]="tamara", *p;
for(i=0; i<strlen(c); i=i+1)
{ p=strchr(c, 'a');
  cout<<p-c; }
```

24. Ce se va afișa în urma executării secvenței alăturate, în care variabila c memorează un șir cu cel mult 20 de caractere, iar variabila i este de tip întreg?

```
char c[]="tamara";
for(i=0; i<3; i++)
c[i]=c[i+1];
```



```
cout<<c;
```

25. Știind că în urma executării secvenței alăturate s-a afișat succesiunea de caractere EXAMEN, care este șirul de caractere memorat de variabila s?

```
x=strlen(s);  
for (i=0;i<x/2;i++)  
cout<<s[i]<<s[x-i-1];
```

26. Considerăm că variabila s memorează șirul de caractere examen. Care va fi valoarea lui s după executarea instrucțiunilor scrise alăturat?

```
s[0]= 'E';  
s[strlen(s)-1]= 'A';  
s[strlen(s)/2-1]= 'N';  
s[strlen(s)/2]= 'M';
```

27. Ce se afișează pe ecran în urma executării secvenței de program scrisă alăturat, în care i este o variabilă de tip char?

```
for (i='a';i<='z';i++)  
if (strchr("info",i))  
cout<<i;
```

28. Ce se afișează pe ecran în urma executării secvenței de program alăturate, unde a este o variabilă de tip șir de caractere?

```
strcpy(a,"informatica");  
strcpy(a+2,a+5);  
cout<<a;
```

29. Ce valoare se va afișa pe ecran în urma executării secvenței de program alăturate, știind că variabila a este de tip șir de caractere, iar i este o variabilă de tip întreg?

```
strcpy(a,"info");  
for(i=0;i<strlen(a);i++)  
a[i]=a[i]+1;  
cout<<a;
```

30. Variabila s reține șirul de caractere bacalaureat. Ce se afișează la executarea instrucțiunii de mai jos?

```
cout<<strchr(s, 'a');
```

31. În secvența alăturată, variabilele s1, s2 și s3 rețin șiruri de caractere. După executarea acesteia, variabila întregă val primește valoarea 1 dacă:

```
if(!(strcmp(s1,s2) || strcmp(s1,s3)))  
    val=1;  
    else val=2;
```

a. s1, s2, s3 rețin șiruri identice de caractere

b. s1, s2, s3 rețin șiruri de caractere ordonate lexicografic

c. s1, s2, s3 rețin șiruri de caractere de lungimi diferite

d. s1 este obținut prin concatenarea șirurilor reținute în s2 și s3

32. În secvența de program următoare, variabila s memorează un șir de caractere, iar variabila i este de tip întreg. Ce valoare se va afișa în urma executării secvenței?

```
strcpy(s, "bacalaureat");  
i=strchr(s, 'a')-s;  
cout<<i+1;
```

33. În secvența alăturată, variabila x memorează un șir de caractere, iar toate celelalte variabile sunt de tip întreg. Ce valori au variabilele k1 și k2 după executarea secvenței de instrucțiuni alăturate?

```
strcpy(x, "bac2009");  
k1=strlen(x);  
k2=0;  
for (i=0; i<strlen(x); i++)  
    if ( x[i]>='0' && x[i]<='9' )  
        k2=k2+1;
```

34. Variabila s memorează un șir de caractere. Care dintre următoarele expresii C++ este nenulă dacă și numai dacă lungimea efectivă a șirului este strict mai mică decât 10?

- a. `strlen(s)<10`
- b. `strlen(s,10)<0`
- c. `leng(s)<10`
- d. `s-'0'<10`

35. Variabila s memorează un șir de caractere. Care dintre următoarele expresii C++ este nenulă dacă și numai dacă lungimea efectivă a șirului este un număr par?

- a. `s-2==0`
- b. `strlen(s,2)=0`
- c. `leng(s)%2`
- d. `strlen(s)%2==0;`

36. Care vor fi valorile afișate după executarea secvenței de program alăturate?

```
char s1[20]="algoritm",
s2[20]="bioritm",s3[20]="ritm";
if (strlen(s1)< strlen(s2))
strcat(s3,s1);
else
strcat(s3,s2);
cout<<s1<<' '<<s2<<' '<<s3;
```

- a. algoritmritm bioritm ritm
- b. algoritm bioritm ritmalgoritm
- c. algoritm bioritm ritmbioritm
- d. algoritm bioritmritm ritm

37. În secvența alăturată, fiecare dintre variabilele x și s sunt de tipul șir de caracter, iar i este de tip întreg. Dacă variabilele x și s memorează inițial șirul absolut, ce se va memora în variabila x în urma executării secvenței alăturate?

```
for(i=0; i<strlen(s); i++)
if(strcmp(x, s+i)<0)
strcpy(x,s+i);
```

38. Ce se va afișa în urma executării secvenței de program de mai jos dacă variabila x memorează cuvântul bacalaureat, iar variabila y memorează cuvântul banal?

```
if(strcmp(x, y) > 0) cout << x;
else
if(strcmp(x,y) < 0) cout << y;
else cout << "imposibil";
```

39. Ce se va afișa în urma executării secvenței de program alăturate știind că i este o variabilă de tip întreg, iar variabila x este de tip șir de caractere?

```
strcpy(x, "ExAMeNe NaTiOnALe");
for(i = 0; i < strlen(x); i++)
if(x[i] >= 'A' && x[i] <='N')
x[i] = x[i] + 'a' - 'A';
cout << x;
```

40. Un șir de caractere se numește palindrom dacă șirul citit de la stânga la dreapta este identic cu șirul citit de la dreapta spre stânga. Care dintre următoarele expresii C/C++ are valoarea 1 dacă și numai dacă șirul de caractere memorat în variabila s, având exact 3 caractere, este palindrom?

- a. $s[0]==s[1]$ b. $s[1]==s[2]$ c. $s[0]==s[2]$
d. $s[1]==s[3]$

41. Știind că variabila i este de tip întreg și variabila s reține un șir de caractere, ce se va afișa la executarea secvenței alăturate?

```
strcpy(s, "bac2009");
for(i=0; i<strlen(s); i++)
if(s[i]<'0' || s[i]>'9')
cout<<s[i];
```

42. Variabila s reține un șir de caractere format din cel puțin 2 și cel mult 30 de litere mici ale alfabetului englez. Scrieți secvența de program C/C++ care afișează pe ecran primul și ultimul caracter al șirului s.

43. Ce valoare are variabila s de tip șir de caractere după executarea instrucțiunilor de mai jos?

```
strncpy(s, strstr("examen", "am"), 4); s[4]='\0';
```

44. Ce valoare are variabila s de tip șir de caractere după executarea instrucțiunilor de mai jos?

```
strncpy(s, strstr("Informatica", "form"), strlen("BAC08")); s[5]='\0';
```

45. Ce valoare are variabila s de tip șir de caractere după executarea instrucțiunii de mai jos?

```
s=strcat(strstr("bacalaureat", "bac")+  
strlen("2009"), "09");
```

46. Ce valoare are variabila s de tip șir de caractere după executarea instrucțiunilor de mai jos?

```
strncpy(s, strstr("informatica", "form"), strlen("BAC009"));  
s[6]='\0';
```

47. Ce memorează variabila s, de tip șir de caractere, după executarea instrucțiunilor de mai jos?

```
strncpy(s, "informatica", strlen("2009"));  
s[strlen("2009")]='\0';  
strcat(s, "BAC");
```

48. Ce se va afișa în urma executării secvenței alăturate de program, în care variabila c memorează un șir cu cel mult 20 de caractere, iar variabila i este de tip întreg?

```
char c[]="abracadabra";  
i=6;  
cout<<c[i]<<c[i+1] <<endl;  
while (i>=0)  
{cout<<c[i];  
i=i-1;}
```

49. Variabila s este de tip șir de caractere, iar variabilele c1 și c2 sunt de tip char. Care

expresie are valoarea 1 dacă și numai dacă șirul de caractere s conține caracterele memorate de variabilele c1 și c2?

- a. `strstr(s, c1+c2) != 0`
- b. `strchr(s, c1) != 0 || strchr(s, c2) != 0`
- c. `strchr(strchr(s, c1), c2) != 0`
- d. `strchr(s, c1)*strchr(s, c2) != 0`

50. Se consideră variabila s care memorează șirul de caractere CARACATITA. Ce valoare va avea s după executarea instrucțiunii de mai jos?

```
strcpy(s, strstr(s, "TI"));
```

51. Ce se va afișa în urma executării secvenței de program alăturate, considerând că a este o variabilă de tip șir de caractere, iar i o variabilă de tip întreg?

```
char a[10]="Examen";  
for (i=0; i<=2; i++)  
    strcpy(a+i, a+i+1);  
cout<<a;
```

52. Considerăm variabila x care memorează șirul de caractere ABAC. Care dintre următoarele instrucțiuni conduc la afișarea caracterului B?

- a. `cout<<x[strlen(x)-3];`
- b. `cout<<x[strlen(x)-1];`
- c. `cout<<x[2];`
- d. `cout<<x[strlen(x)];`

53. Se consideră declararea `char e[20]="51+73";` Care este șirul memorat de variabila e după executarea instrucțiunii de mai jos?

```
strcpy(e, strchr(e, '+' )+2);
```

Probleme rezolvate:

Problema 1:

Să se scrie o funcție care să returneze lungimea unui șir de caractere, transmis ca parametru(fără a folosi funcția sistem).

```
#include<iostream.h>
int lungime(char s[30])
{
    int i;
    i=0;
    while(s[i]!=0) i++;
    return i;
}
void main(void)
{
    char s[30];
    int l;
    cout<<"Dati sirul de caractere „ „ ;
    cin>>s;
    l=lungime(s);
    cout<<"Numarul de caractere ale sirului
    este "<<l;
}
```

Problema 2:

Scrieți un program care citește un șir de caractere și transformă șirul în șir cu litere mici.

```
#include<iostream.h>
void litere_mici(char s[30])
{
    int i;
    i=0;
    while(s[i]!=0)
    {
        if( (s[i]>=65) && (s[i]<=90) )
            s[i]=s[i]+32;
        i++;
    }
}
```

```

return;
}
void main(void)
{
char s[30];
int l;
cout<<"Dati sirul de caractere
";cin>>s;
litere_mici(s);
cout<<"Sirul transformat in litere mici
este "<<s;
}

```

Problema 3:

Se citește un șir de caractere ce nu conține caractere albe.Să se afișeze toate cuvintele obținute prin eliminarea unui singur caracter

Exemplu :citim asde;se va afisa:sde,ade,ase,asd

```

#include<iostream.h>
#include<string.h>
void main()
{
char s[20];
int i,j;
cin>>s;
for(i=0;i<strlen(s);i++ )
{
for(j= 0;j<strlen(s);j++)
if(i!=j)
cout<<s[j];
cout<<endl;
}
}

```


Problema 4.

Se citește un șir de caractere ce nu conține caractere albe. Să se elimine toate consoanele și să se afișeze șirul obținut.

```
#include<iostream.h>
#include<string.h>
void main()
{
char s[20],vocale[]="aeiouAEIOU";
int gasit=1;;
cin>>s;
int i;
while(gasit==1)
for(i=0;i<strlen(s);i++)
{
gasit=0;
if (strchr(vocale,s[i])==0)
{
strcpy(s+i,s+i+1);
gasit=1;
}
}
cout<<s;
}
```

Problema 5.

Se citește o propoziție. Să se transforme toate literele mici în litere mari.

```
#include<iostream.h>
#include<string.h>

void main()
{
char s[20];
```

```
cin.get(s,20);
strupr(s);
cout<<s;
}
```

Problema 6.

Se citește un șir de caractere ce nu conține caractere albe. Să se elimine vocalele din șir.

```
#include<iostream.h>
#include<string.h>
void main()
{
char s[20],vocale[]="aeiouAEIOU";
int gasit=1;
cin>>s;
int i;
while(gasit)
for(i=0;i<strlen(s);i++)
{
gasit=0;
if (strchr(vocale,s[i])!=0)
{strcpy(s+i,s+i+1); gasit=1;}
}

cout<<s;
}
```

Problema 7.

Se citește un cuvânt fără caractere albe. Să se înlocuiască fiecare vocală mică din cuvânt cu vocala mare corespunzătoare.

```
#include<iostream.h>
#include<string.h>
```

```

void main()
{ char vocale[]="aeiou";
  char s[20];
  int i;
  cin>>s;
  for(i=0;i<strlen(s);i++)
    if(strchr(vocale,s[i])!=0) s[i]=s[i]+'A'-'
'a'; //sau s[i]=s[i]-32;
  cout<<s;
}

```

Problema 8.

Se citește un șir de caractere care nu conține caractere albe.Să se verifice dacă șirul e alcătuit exclusiv din caractere nenumerice

```

#include<iostream.h>
#include<string.h>
void main()
{
char cuvant[100],cifre[]="0123456789";
cout<<"cuvant:";cin>>cuvant;
if(strcspn(cifre,cuvant)==10)
cout<<"caractere nenumerice";
else cout<<"caractere numerice";
}

```

Problema 9.

Se citește un șir de caractere care nu conține caractere albe.Să se verifice dacă șirul este alcătuit exclusiv din caractere numerice.

```

#include<iostream.h>
#include<string.h>
void main()
{

```

```

char cuvant[100], cifre[]="0123456789";
cout<<"cuvant:"; cin>>cuvant;
if(strspn(cuvant, cifre)==strlen(cuvant))
cout<<"numeric";
else cout<<"nenumeric";

```

Problema 10.

Se citesc n cuvinte. Să se sorteze alfabetic cuvintele citite.

```

#include<iostream.h>
#include<string.h>
void main()
{
char cuvinte[10][25], aux[25];
int n, i, gasit;
cout<<"n="; cin>>n;
for(i=0; i<n; i++)
{cout<<"cuvant:"; cin>>cuvinte[i];}
do
{
gasit=0;
for(i=0; i<n-1; i++)
if(strcmp(cuvinte[i], cuvinte[i+1])>0)
{
strcpy(aux, cuvinte[i]);
strcpy(cuvinte[i], cuvinte[i+1]);
strcpy(cuvinte[i+1], aux);
gasit=1;
}
}while(gasit);
for(i=0; i<n; i++) cout<<cuvinte[i]<<" ";
}

```

Problema 11.

Să se citească de la tastatura un nume și să se determine dacă este nume de fata sau de baiat (toate numele de fată, cu exceptia numelui "Carmen" se termină cu o vocală):

```

#include <stdio.h>
#include <string.h>
#include <ctype.h> /* pentru toupper */
int main (void)
{
char nume [80];
int n;
char c;
do
{
cout<<"Introduceti numele: ";
cin>>nume;
n = strlen (nume);
}while (n == 0);
/* nu permitem introducerea unui sir vid
*/
c = toupper (nume [n - 1]);
if ((c=='A')|| (c=='E') || (c=='I') ||
(c=='O'))
cout<<"Nume de fata";
else
cout<<"Nume de baiat ";
return 0;
}

```

Problema 12.

Să se citească de la tastatură un șir de caractere și să se afișeze șirul în ordine inversă:

```

#include <stdio.h>
#include <string.h>
int main (void)
{
char sir [80];
int k, n;
cout<<"Introduceti sirul: ";
cin>>sir;
n = strlen (sir);
for (k = n - 1; k >= 0; k--)

```

```
cout<< sir [k];  
return 0;}
```

Problema 13.

Să se citească un șir de caractere reprezentând un număr în baza 2 și să se afișeze valoarea acestuia în baza 10.

```
#include <stdio.h>  
#include <string.h>  
int main (void)  
{  
char sir [80];  
int k, n, val, p, cifra, valid;  
do  
{  
do  
{  
cout<<"Introduceti sirul: ";  
cin>>sir;  
n = strlen (sir);  
}while (n == 0);  
valid = 1;  
for (k = 0; k < n; k++)  
if ((sir [k] != '0') && (sir [k] !=  
'1'))  
valid = 0;  
if (!valid)  
cout<<"Sirul introdus nu este numar in  
baza 2 ! ";  
}while (!valid);  
p = 1; /* puterea lui 2 */  
val = 0; /* valoarea numarului */  
for (k = n - 1; k >= 0; k--)  
{  
cifra = sir [k] - '0';  
val = val + cifra * p;  
p = p * 2;  
}
```

```
Cout<<"Numarul"<<"in baza 10 este:  
„<<sir<<val;  
return 0;}
```

Probleme propuse:

1. De la tastatură se citește un cuvânt având maximum 25 de caractere. Să se afișeze toate sufixele cuvântului, fiecare pe câte o linie.
2. De la tastatură se citește un cuvânt având maximum 25 de caractere. Să se afișeze toate prefixele cuvântului care încep cu o vocală , fiecare pe câte o linie.
3. Se citește de la tastatură un text. Să se determine numărul caracterelor de tip cifră din text.
4. Se citește de la tastatură un text. Să se determine numărul caracterelor de spațiu din text.
5. Pentru un text citit să se afișeze același text, dar toate caracterele alfabetice să fie mici, iar cele numerice să se elimine.
6. Se citește de la tastatură un text. Să se afișeze numărul de consoane din text, precum și consoanele distincte.
7. De la tastatură se citește un text în care cuvintele sunt separate prin unul sau mai mulți separatori (., ;' ?!:]. Să se afișeze pe câte o linie cuvintele care conțin cel puțin 2 vocale distincte.
8. Se dau două texte citite de la tastatură. Să se afișeze caracterele de tip cifră comune celor două texte.
9. Se citește de la tastatură un text. Să se afișeze toate caracterele alfabetice, mici sau mari, care nu se regăsesc în text.
10. De la tastatură se citesc două cuvinte. Să se verifice dacă cuvintele sunt formate din aceleași caractere, iar în caz afirmativ să se afișeze caracterele corespunzătoare.
11. Din fișierul '*intrare.in*' se citesc cuvinte formate numai din caractere litere mici. Fiecare cuvânt se găsește pe câte o linie. Să se afișeze caracterele comune cuvintelor din fișier, caractere care se vor afișa în fișierul '*iesire.out*'.
12. De la tastatură se citește un text în care cuvintele sunt separate prin unul sau mai mulți separatori (., ;' ?!:]. Să se afișeze pe câte o linie cuvintele care conțin cel mult 3 vocale distincte.
13. De la tastatură se citește un text în care cuvintele sunt separate prin unul sau mai mulți separatori (., ;' ?!:]. Să se afișeze pe câte o linie cuvintele care conțin 2 vocale distincte.

14. De la tastatură se citește un text în care cuvintele sunt separate prin unul sau mai mulți separatori (., ;' ?!:]. Să se afișeze pe câte o linie cuvintele care conțin cel puțin 3 vocale.
15. Să se scrie un program C++ care citește de la tastatură un cuvânt format din cel mult 20 de caractere, doar litere mici ale alfabetului englez. Programul determină transformarea cuvântului citit prin înlocuirea fiecărei vocale a cuvântului, cu un șir format din două caractere și anume vocala respectivă urmată de litera mare corespunzătoare, restul literelor nemodificându-se, ca în exemplu. Programul afișează pe ecran cuvântul obținut, pe o singură linie. Se consideră vocale literele din mulțimea {a,e,i,o,u}.

*Exemplu: pentru cuvântul bacalaureat se va afișa pe ecran:
baAcaAlaAuUreEaAt*

(Bacalaureat 2009)

13. Se consideră un text cu maximum 255 de caractere în care cuvintele sunt separate prin unul sau mai multe spații. Primul caracter din textul citit este o literă, iar cuvintele sunt formate numai din litere mici ale alfabetului englez. Scrieți un program C++ care citește de la tastatură textul și îl transformă, înlocuind prima literă a fiecărui cuvânt cu litera mare corespunzătoare, restul caracterelor rămânând nemodificate. Textul astfel transformat va fi afișat pe ecran.

Exemplu: dacă de la tastatură se introduce textul: mare frig rosu se va afișa pe ecran: Mare Frig Rosu

(Bacalaureat 2009)

14. Se consideră un text cu maximum 255 de caractere, format din litere mici ale alfabetului englez și spații. Textul conține cel puțin o consoană. Scrieți un program C++ care citește de la tastatură textul și apoi determină transformarea acestuia, eliminând numai ultima consoană care apare în text, ca în exemplu. Programul va afișa pe ecran textul obținut.

Exemplu: dacă de la tastatură se introduce textul: mare frig saci pe ecran se va afișa: mare frig sai

(Bacalaureat 2009)

15. Se consideră un text alcătuit din cel mult 250 de caractere, în care cuvintele sunt formate doar din litere mici ale alfabetului englez și sunt separate prin unul sau mai multe caractere *.

Scrieți un program C++ care citește de la tastatură textul și afișează pe ecran, pe câte o linie, toate secvențele formate din câte două litere identice, ca în exemplu.

Exemplu: dacă textul citit este: se afișează perechile alăturate :ii ii oo

(Bacalaureat 2009)

15. Scrieți un program C++ care citește de la tastatură două șiruri de caractere formate din maximum 100 litere mici ale alfabetului englez și afișează pe ecran cel mai lung sufix comun al celor două șiruri de caractere. Dacă cele două șiruri nu au niciun sufix comun, atunci programul va afișa pe ecran mesajul NU EXISTĂ.

Exemplu: pentru șirurile marina și elena se va afișa na

(Bacalaureat 2009)

16. Șirul de caractere s2 este “clona” șirului de caractere s1 dacă se poate obține din s1 prin eliminarea tuturor aparițiilor unei singure vocale. Se consideră vocală orice literă din mulțimea {a,e,i,o,u}. Scrieți programul C/C++ care citește de la tastatură un cuvânt format din cel mult 20 litere mici ale alfabetului englez și afișează pe ecran (dacă există), toate “clonele” acestui cuvânt, fiecare pe câte o linie a ecranului.

Exemplu: pentru cuvântul informatica se afișează, nu neapărat în această ordine, “clonele” scrise alăturat.

nformatca

infrmatica

informtic

(Bacalaureat 2009)

17. Să se scrie un program C++ care să citească de la tastatură n șiruri de caractere ($n < 10$, introduse de la tastatură) și afișează cel mai mare dintre șiruri din punct de vedere alfabetic.

Indicatie: Un tablou de siruri de caractere se declara ca un tablou bidimensional:

char tab [10][80];

Variabila tab are alocat spatiu pentru 10 siruri de caractere, fiecare sir având lungimea maxima de 79 de caractere + terminatorul de sir. Primul sir este tab [0], ultimul este tab [9].

18. Să se scrie un program C++ care realizează următoarele:
a) șterge dintr-un șir de caractere un subșir specificat prin poziție și lungime.

- b) inserează într-un șir începând cu o poziție dată un al șir.
- c) citește două cuvinte și înlocuiește într-un text introdus de la tastatură toate aparițiile primului cuvânt prin cel de-al doilea.
19. Să se citească un text de la tastatură care se încheie cu ‘.’. Să se afișeze cel mai lung cuvânt din text (dacă sunt mai multe cu aceeași lungime maximă se va afișa doar unul dintre ele).
20. Să se implementeze funcțiile *strlen*, *strcpy*, *strcmp*, *strstr*, *strcat*.
21. Să se scrie un program în C care să permită gestiunea cartilor dintr-o bibliotecă, prin intermediul unui meniu interactiv cu următoarele opțiuni:
- 1) Adăugare carte (titlu și autor)
 - 2) Căutare după titlu
 - 3) Căutare după autor
 - 4) Ștergere carte
 - 5) Ieșire
22. Să se determine de câte ori se găsește un cuvânt într-un text.
23. Codificați un text astfel încât litera a să devină c, b să devină e și a.m.d.
24. Să se sorteze alfabetic un șir de cuvinte (eventual, fără a distinge literele mici de cele mari).
25. Codificarea pășărească a unui cuvânt (după fiecare vocală, se pune litera p urmată de acea vocală). Ex : informatică → ipinfopormapatipicapa
26. Se citește un cuvânt. Să se afișeze grupurile de cuvinte care rimează (au ultimele 2 caractere identice).
27. Să se despartă un text în cuvinte și să se afișeze cuvintele separate. Să se afișeze cuvântul de lungime maximă.
28. Să se verifice dacă două cuvinte sunt sau nu anagrame.
29. Să se numere aparițiile unui cuvânt într-un text.
30. Se citește un număr real de la tastatură. Să se trunchieze astfel încât cifrele rămase să formeze o secvență monotonă. Ex. 34.59483 → 34.59 ; 24.1276 → 24
31. Se citește un șir de caractere alfanumerice. Considerăm că literele sunt separatorii numerelor. Afișați datele de tip numeric preluate în ordine din șirul citit. Numerele vor fi scrise câte unul pe o linie.

Ex.

<i>in.txt</i>	<i>out.txt</i>
a23sc345ss5e	23
	345
	5

32. În directorul curent se află fișierul cuvinte.txt care conține mai multe linii de text formate din cuvinte separate de câte un spațiu. Să se afișeze cuvintele care au cel puțin 3 consoane sau 3 vocale consecutive.
33. Se citește un șir de caractere. Să se afișeze șirul oglindit din care lipsesc vocalele.
34. Se dă un text de maxim 30 de caractere. Să se listeze toate cuvintele de două caractere din acest text.
35. Se introduc de la tastatură cuvinte până la citirea cuvântului stop. Afișați pe ecran cuvintele mai mari în sens lexicografic decât primul citit.
36. Se dau două texte. Să se stabilească o vocală comună celor două texte, care apare de cele mai puține ori.
37. Dintr-un fișier se citește un text. Textul conține cuvinte separate printr-un spațiu. Să se determine câte cuvinte conține textul.
38. Dintr-un fișier se citește un text. Textul conține cuvinte separate printr-un spațiu. Se va genera un nou text care va conține cuvintele ordonate alfabetic.
39. Dintr-un fișier se citește un text. Textul conține cuvinte separate printr-un spațiu. Să se scrie într-un alt fișier, pe linii separate, fiecare cuvânt care apare în text urmat de un număr care va reprezenta de câte ori apare cuvântul în text. Să se determine cuvântul care apare de cele mai multe ori.
40. Dintr-un fișier se citește un text. Textul conține cuvinte separate printr-un spațiu. Într-un alt fișier se va scrie pe linii separate fiecare cuvânt și liniile pe care apare.
41. Dintr-un fișier se citește un text. Textul conține cuvinte separate printr-un spațiu sau mai multe. Se va genera un nou fișier care va conține textul inițial având spațiile de prisos eliminate (între cuvinte va rămâne numai câte un spațiu).
42. Se citesc de la tastatură elementele unei matrici de caractere (nr. linii=nr. coloane), $A(N \times N)$, $N \leq 10$.
 - a. Să se afișeze matricea A ;

- b. Să se formeze și să se afișeze cuvântul format din caracterele pe pe diagonala principala a matricii A;
 - c. Să se calculeze și să se afișeze numărul de litere mari, litere mici și cifre din matrice;
 - d. Să se afișeze cuvântul format din caracterele de pe diagonala secundară;
 - e. Să se afișeze procentul literelor mari, al literelor mici și al cifrelor de pe cele 2 diagonale;
 - f. Să se afișeze caracterele comune aflate pe liniile p și q ($p, q < N$, p și q citite de la tastatură);
 - g. Să se afișeze în ordine alfabetică, crescătoare, literele mari aflate pe coloanele impare.
43. Simulați comanda REPLACE astfel încât într-un text veți înlocui un caracter x citit de la tastatură cu un alt caracter y citit de la tastatură. Se ignoră sau nu diferența de cheie după opțiunea utilizatorului.
 44. Simulați comanda REPLACE astfel încât într-un text veți înlocui un șir x citit de la tastatură cu un alt caracter șir y citit de la tastatură. Se ignoră sau nu diferența de cheie după opțiunea utilizatorului.
 45. Se citește de la tastatură un cuvânt. Să se stabilească dacă el conține două litere alăturate identice, afișându-se un mesaj corespunzător.
 46. Dintr-un fișier se citesc numele a n persoane. Să se modifice conținutul fișierului astfel încât toate numele să fie scrise astfel: prima literă mare și restul litere mici.
 47. Se citesc n șiruri. Pentru fiecare șir se va determina suma codurilor ASCII.
 48. Într-un fișier sunt scrise cuvinte pe linii separate. Să se afișeze cuvintele care conțin majuscule.
 49. Într-un fișier sunt scrise pe rânduri diferite numele a n copii. Să se modifice conținutul fișierului astfel încât să conțină numele ordonate crescător.
 50. Să se afișeze vocalele unui cuvânt.
 51. Să se afișeze cuvintele care încep și se termină cu consoana, (vocala) etc.
 52. Să se despartă un text în cuvinte și să se afișeze cuvintele separate. Să se afișeze cuvântul de lungime maximă.

53. Într-un text există un cuvânt. Codificați/decodificați cuvântul după un algoritm generat de voi.
54. Aceeași problemă pentru un text.
55. Se dau două texte. Să se stabilească o vocală comună celor două texte, care apare de cele mai puține ori.
56. Dintr-un fișier se citește un text. Textul conține cuvinte separate printr-un spațiu. Să se determine câte cuvinte conține textul.
57. Se citesc n cuvinte. Să se afișeze perechile de 2 cuvinte care rimează.
58. Aceeași problemă, numai ca se vor afișa grupurile de cuvinte care rimează.
59. Se citesc cuvinte până la citirea cuvântului "stop". Să se afișeze câte dintre cuvintele citite sunt egale cu primul cuvânt citit.
60. Se citește un text. Textul conține cuvinte separate printr-un spațiu. Se va genera un nou text care va conține cuvintele ordonate alfabetic.
61. Se citește un text. Textul conține cuvinte separate printr-un spațiu. Să se scrie, pe linii separate, fiecare cuvânt care apare în text urmat de un număr care va reprezenta de câte ori apare cuvântul în text. Să se determine cuvântul care apare de cele mai multe ori.
62. Se citește un text. Textul conține cuvinte separate printr-un spațiu. Să se scrie, pe linii separate, fiecare cuvânt și liniile pe care apare.
63. Se citește un text care conține cuvinte separate printr-un spațiu sau mai multe. Se va genera un nou text care va conține textul inițial având spațiile de prisos eliminate (între cuvinte va rămâne numai câte un spațiu).
64. Simulați scrierea unei parole într-un fișier. La tastarea parolei pe ecran în locul fiecărui caracter se va scrie caracterul '*'. Eventual realizați și încriptarea parolei înainte de a fi scrisă într-un fișier.
65. Fie un șir de forma: cifra-literă, cifra literă ...etc. Ex 2a4b5c. Să se genereze un astfel de șir: aabbbbcccc.
66. Fie un șir format din replicarea de un număr de ori a unui subșir redundant. Să se determine cea mai scurtă secvență redundantă. Se va afișa subșirul redundant și numărul său de apariții. Ex pt: aabcaabcaabcaabc se va afișa: aabc și nu aabcaabc. Numărul de apariții este 4 și nu 2.

CAPITOLUL II ÎNREGISTRĂRI

Itemii următori sunt preluați din variantele de bacalaureat 2009.

1. În declararea alăturată, câmpurile x și y ale înregistrării pot memora coordonatele carteziene ale unui punct din planul xOy . Scrieți o secvență de instrucțiuni prin executarea căreia se memorează în variabila C coordonatele mijlocului segmentului cu capetele în punctele ale căror coordonate sunt memorate în variabilele A și B .

```
struct punct
{
float x, y;
}A, B, C;
```

2. Variabila t , declarată alăturat, memorează în câmpurile a , b și c lungimile laturilor unui triunghi. Care dintre următoarele instrucțiuni atribuie câmpului p al variabilei t valoarea perimetrului triunghiului respectiv?

```
struct triunghi
{
float a, b, c, p;
}t;
```

3. În declararea alăturată, câmpurile x și y ale înregistrării pot memora numărătorul, respectiv numitorul unei fracții. Scrieți secvența de instrucțiuni prin executarea căreia se construiește în variabila f o fracție obținută prin însumarea fracțiilor memorate în variabilele $f1$ și $f2$.

```
struct fractie
{
int x, y;
}f, f1, f2;
```

4. Variabila d , declarată alăturat, memorează în câmpurile a și b lățimea și, respectiv, lungimea unui dreptunghi. Care dintre următoarele instrucțiuni atribuie câmpului $aria$ al variabilei d valoarea ariei dreptunghiului respectiv?

```
struct dreptunghi
```

```
{
float a,b,aria;
}d;
```

5. Considerând declararea alăturată, care dintre următoarele secvențe realizează în mod corect citirea de la tastatură a valorilor celor două câmpuri ale variabilei x?

```
struct {
int a;
int b;} x;
```

a. cin>>x.a>>x.b;
b. cin>>a.x>>b.x;
c. cin>>x;
d. cin>>a->x>>b->x;

6. Care dintre următoarele variante reprezintă o declarație corectă pentru o variabilă x care memorează simultan vârsta în ani împliniți și media la bacalaureat a unui elev?

a. struct {float media;
int varsta;} x;
b. struct x {float media;
int varsta};
c. float x.media;
int x.varsta;
d. struct elev {float x.media;
int x.varsta};

7. Care dintre următoarele variante reprezintă o declarație corectă pentru o variabilă x care memorează simultan codul de identificare al unui candidat la un examen, exprimat printr-un număr natural de cel mult 4 cifre și media obținută de acesta la examen, exprimată printr-un număr real?

a. struct x { int cod;
float media};
b. struct {int cod;
float media;} x;
c. int x.cod ;
float x.media;
d. struct candidat {int x.cod;
float x.media};

8. Care dintre următoarele variante reprezintă o declarație corectă pentru o variabilă x care memorează simultan coordonatele reale (abscisa și ordonata) ale unui punct în planul xOy ?

- a. `struct punct{float ox,oy;} x;`
- b. `char x[2];`
- c. `struct x{float ox,oy;};`
- d. `float x;`

9. Fiecare dintre variabilele a și b , declarate alăturat, memorează simultan coordonatele reale ale câte unui punct în planul xOy .

```
struct punct{
    float x,y;}a,b;
```

Completați punctele de suspensie din secvența următoare de program, astfel încât, în urma executării ei, să se afișeze cuvântul DA dacă dreapta, determinată de punctele cu coordonatele memorate în variabilele a și b , este paralelă cu axa Ox , respectiv cuvântul NU în caz contrar.

```
if (...) cout<<"DA";
else cout<<"NU"
```

10. Se consideră declarațiile de mai jos, în care variabila ev memorează date despre un anumit elev. Scrieți instrucțiunea C/C++ prin care se inițializează anul nașterii acestui elev cu valoarea 1990.

```
struct data{
    int zi;
    int luna;
    int an;
};
struct elev {
    char nume[30];
    struct data data_nasterii;
    float media;
}ev;
```

Probleme rezolvate

1. Se citesc n fracții dintr-un fișier text. Să se afișeze fracțiile reducibile ordonate crescător.

Ex: Dacă se citesc următoarele 4 fracții: $3/6$ $2/8$ $3/7$ $2/20$ se vor afișa $2/20$ $2/8$ $2/6$

```
#include<fstream.h>
struct fractie {int nu,nr;
```

```

        };
ifstream f("fractii.in");
void citire(fractie &e)
{f>>e.nr>>e.nu;
}
void afisare(fractie e)
{cout<<e.nr<<"/"<<e.nu<<" ";
}
void ordonare(int n, fractie e[20])
{int i, gata;
  fractie aux;
  do{gata=1;
    for(i=1; i<n; i++)
      if
(e[i].nr*e[i+1].nu>e[i].nu*e[i+1].nr)
        {aux=e[i];
          e[i]=e[i+1];
          e[i+1]=aux;
          gata=0;
        }
    }while (!gata);
}
int cmmdc(int a, int b)
{while(a!=b)
  if (a>b) a=a-b;
  else b=b-a;
  return a;
}
int main()
{int i, n;
  fractie a[20];
  f>>n;
  for (i=1; i<=n; i++) citire(a[i]);
  ordonare(n, a);
  for (i=1; i<=n; i++)
    if(cmmdc(a[i].nr, a[i].nu)!=1)
      afisare (a[i]);
  return 0;
}

```

```
}
```

2. Despre fiecare dintre cei n elevi dintr-o clasă se citesc următoarele date:

- numele (șir de caractere de lungime maximă 20) ;
- numele (șir de caractere de lungime maximă 20) ;
- media notelor în oral la informatică;
- nota în teza la informatică .

Să se afișeze ordonați alfabetic elevii corigenți la informatică. Pentru fiecare elev corigent se vor afișa: numele, prenumele și media la informatică.

```
#include<fstream.h>
#include<string.h>
struct elev {char nume[30];
             char pren[30];
             float mo;
             int teza;
             float media;
             };
ifstream f("elev.in");
elev c[100];
void citire(elev &e)
{f>>e.nume;
 f>>e.pren;
 f>>e.mo;
 f>>e.teza;
 e.media=(e.mo*3+e.teza)/4;
}
void afisare(elev e)
{cout<<e.nume<<" "<<e.pren<<"
"<<e.media;
 cout<<endl;
}
void ordonare(int n,elev c[100])
{int i,gata;
 elev aux;
 do {gata=1;
 for (i=1;i<n;i++)
 if (strcmp(c[i].nume,c[i+1].nume)>0)
```

```

        {aux=c[i];
          c[i]=c[i+1];
          c[i+1]=aux;
          gata=0;
        }
      }while (!gata);
    }
int main ()
{ int n,i;
  f>>n;
  for (i=1;i<=n;i++) citire(c[i]);
  ordonare (n,c);
  for (i=1;i<=n;i++)
    if (c[i].media<4.5)
      afisare(c[i]);
  return 0;
}

```

3. În fișierul *sarituri.in* sunt notate numele, prenumele și lungimea săriturii pentru cei n elevi dintr-o clasă. Numele și prenumele sunt șiruri de caractere, iar lungimea săriturii este număr real. Să se construiască două fișiere text *baieti.out* și *fete.out* în care să se afișeze baietii și respective fetele din clasă, ordonați/ordonate descrescător după lungimea săriturii. Se consideră că toate fetele din clasă au prenumele terminat cu litera a.

```

#include<fstream.h>
#include<string.h>
struct elev
{
    char nume[40],prenume[40];
    float lung;
};
elev a[100];
ifstream f("sarituri.in");
ofstream g1("baieti.out");
ofstream g2("fete.out");
void citire(elev &e)
{

```

```

        f>>e.ume>>" ">>e.prenume>>"
">>e.lung;
    }
void afisare(elev e)
{
    if (e.prenume[strlen(e.prenume)-
1]=='a')
        g2<<e.ume<<" "<<e.prenume<<"
"<<e.lung<<endl;
    else
        g1<<e.ume<<" "<<e.prenume<<"
"<<e.lung<<endl;
}
void ordonare(elev a[100],int n)
{
    int i,gata;
    elev aux;
    do
    { gata=1;
    for(i=1;i<n;i++)
        if(a[i].lung<a[i+1].lung)
        {
            aux=a[i];
            a[i]=a[i+1];
            a[i+1]=aux;
            gata=0;
        }
    }
    while(gata==0);
}
int main()
{
    int n,i;
    f>>n;
    for(i=1;i<=n;i++)
        citire(a[i]);
    ordonare(a,n);
    for(i=1;i<=n;i++)

```

```

        afisare(a[i]);
    f.close();
    g2.close();
    g1.close();
    return 0;
}

```

4. Se citesc două numere raționale x și y . Să se calculeze și să se afișeze suma lor.

```

#include<iostream.h>
struct rational
{ int nr,nu;
}x,y,s;

void main()
{ int a,b;
  cin>>x.nr>>x.nu;
  cin>>y.nr>>y.nu;
  s.nr=x.nr*y.nu+x.nu*y.nr;
  s.nu=x.nu*y.nu;
  a=s.nr;
  b=s.nu;
  while(a!=b)
    if(a>b) a=a-b;
    else b=b-a;
  s.nr=s.nr/a;
  s.nu=s.nu/a;
  cout<<s.nr<<"/"<<s.nu;
}

```

5. Să se definească o structură pentru memorarea unei progresii aritmetice de rație r și termen initial a . Să se afișeze primii 10 termeni ai progresiei date prin a și r .

```

#include<iostream.h>
#include<math.h>
struct progresie_aritmetica
{ double a,r;
}P;

void main()

```

```

    { int i;
      cin>>P.a>>P.r;
      for(i=0;i<=9;i++)
        cout<<P.a+P.r*i<<" ";
    }

```

6. Din fișierul text *clasa.in* se citesc următoarele informații despre fiecare elev din clasă: numele, prenumele și media. Să se afișeze elevii din clasă ordonați descrescător după medie.

```

#include<fstream.h>
ifstream f("clasa.in");
struct elev
{ char nume[20], prenume[20];
  double media;
}C[30];
void main()
{ int i,n;
  f>>n;
  for(i=1;i<=n;i++)

f>>C[i].nume>>C[i].prenume>>C[i].media;
  int gata;
  do{ gata=1;
    for(i=1;i<n;i++)
if(C[i].media<C[i+1].media)
  { elev aux=C[i];
    C[i]=C[i+1];
    C[i+1]=aux;
    gata=0;
  }
  }while(!gata);
  for(i=1;i<=n;i++)
    cout<<C[i].nume<<"
"<<C[i].prenume<<" "<<C[i].media<<endl;
  }

```

7. Din fișierul *f.in* se citesc un număr n și apoi n fracții. Să se scrie în fișierul *f.out* doar fracțiile reducibile din fișierul *f.in*.

```

#include<fstream.h>

```

```

#include<string.h>
struct fractie
{int num,nr;
};
fractie a[100];
ifstream f("f.in");ofstream g("f.out");
void citire(fractie &e)
{ f>>e.num;
  f>>e.nr;
}
void afisare(fractie e)
{ g<<e.num<<"/"<<e.nr;
  g<<endl;
}
int cmmdc(int a,int b)
{
  while(a!=b)
  if(a>b)
  a=a-b;
  else
  b=b-a;
  return a;
}
int main()
{
  int n,i,j;
  f>>n;
  for(i=1;i<=n;i++)
  citire(a[i]);
  for(i=1;i<=n;i++)
  if(cmmdc(a[i].num,a[i].nr)==1)
  {
    for(j=i;j<=n-1;j++)
    a[j]=a[j+1];
    n--;
  }
  for(i=1;i<=n;i++)
  afisare(a[i]);
}

```



```

    f.close();
    g.close();
    return 0;
}

```

8. Să se definească o structură punct pentru memorarea coordonatelor unui punct din plan. Să se declare și să se citească 3 puncte A, B, C. Să se determine dacă punctele sunt coliniare.

```

#include<iostream>
#include<math.h>
using namespace std;
struct punct
{float x,y;
}A,B,C;
float dist(punct A, punct B)
{return sqrt((A.x-B.x)*(A.x-B.x)+(A.y-
B.y)*(A.y-B.y));
}
void citire(punct &A)
{ cin>>A.x>>A.y;
}
int main()
{
    citire(A);
    citire(B);
    citire(C);
    float a,b,c;
    a=dist(B,C);
    b=dist(A,C);
    c=dist(A,B);
    if(a==b+c || c==a+b || b==c+a)
        cout<<"coliniare";
    else cout<<"nu";
}

```

9. Se declară o structură *complex* pentru memorarea numerelor complexe. Să se declare și să se citească o variabilă de tip complex și să se calculeze modulul numărului complex citit.

```

#include<iostream>
#include<cmath>

```

```

using namespace std;

struct Complex{
    float x,y;
};

int main()
{
    Complex c;
    cin >> c.x >> c.y;
    cout << sqrt ( c.x*c.x + c.y*c.y );
    system("pause");
return 0;
}

```

Probleme propuse

1. Să se descompună un număr în factori primi, memorând rezultatul sub forma unui vector de înregistrări: fiecare înregistrare va cuprinde două câmpuri, unul indicând factorul prim, iar celălalt puterea la care apare în descompunere .
2. Dându-se n puncte în plan prin coordonatele (x,y) se cere să se realizeze un program care să afișeze toate punctele aflate în interiorul cercului
3. Se consideră un vector a cuprinzând articole de tip înregistrare:

```

struct Complex{
    float x,y;
};

```

Să se ordoneze vectorul a descrescător după câmpul x, iar în caz de egalitate vectorul se va sorta crescător în funcție de câmpul y.

4. Se consideră un tablou cu înregistrări ce conțin informații referitoare la caracteristicile a n mașini, organizate astfel: marca, an_fab, pret. Știind că o mașină iese din uz după 30 de ani de funcționare, să se realizeze un program care listează toate mărcile mașinilor ieșite din uz, precum și mărcile mașinilor al căror preț se încadrează între două valori introduse de la tastatură (pret1 și pret2).

5. Să se scrie un program care determină zodia pentru o dată de naștere furnizată de la tastatură, sub forma unui număr întreg reprezentând ziua nașterii și a unui șir de caractere pentru luna de naștere. Se folosește o structură de forma:

```
zodii:array[1..12] of
    record
        zi_sf:1..31;
        luna:string;
        nume_zodie:string;
    end;
```

unde `zi_sf`, luna reprezintă ultima dată din zodia respectivă.

CAPITOLUL III SUBPROGRAME

Probleme rezolvate:

1. Să se afișeze toate numerele prime mai mici strict decât o valoare dată n.

```
#include<iostream.h>
#include<math.h>
#include<conio.h>
int prim(int x)
{ int j;
  if (x==0||x==1) return 0;
  for (j=2;j<=floor(sqrt(x));j++)
    if (x%j==0) return 0;
  return 1;
}
void main()
{
  int i,n;
  clrscr();
  cout<<"n=";cin>>n;
  for (i=1;i<n;i++)
    if (prim(i))
      cout<<i<<" ";
  getch();
}
```

2. Afișați toate numerele mai mici sau egale cu n natural pentru care suma cifrelor este pară.

```
#include<iostream.h>
#include<conio.h>
int s_cif(int x)
{ int s=0;
  while (x!=0)
  {
    s+=x%10;
    x=x/10;
  }
}
```

```

    return s;
}
void main()
{
    int n,i;
    clrscr();
    cout<<"n=";
    cin>>n;
    for (i=1;i<=n;i++)
    if (s_cif(i)%2==0) cout<<i<<" ";
    getch();
}

```

3. De la tastatură se citește un număr natural n. Să se determine și să se afișeze numărul obținut prin inversarea cifrelor.

```

#include<iostream.h>
#include<conio.h>
int invers(long int x)
{ long int s=0;
  while (x!=0)
  {
      s=s*10+x%10;
      x=x/10;
  }
  return s;
}
void main()
{ long int n;
  clrscr();
  cout<<"n=";cin>>n;
  cout<<invers(n);
  getch();
}

```

4. Returnați numărul de apariții al unui caracter citit de la tastatură într-un cuvânt.

```

#include<iostream.h>
#include<string.h>
#include<conio.h>
int aparitii(char c,char s[100])
{
    int nr=0,i;
    int m=strlen(s);

```

```

        for(i=0;i<m;i++)
            if (c==s[i]) nr++;
        return nr;
    }
void main()
{
    char s[100],c;
    clrscr();
    cout<<"dati cuvantul =";
    cin>>s;
    cout<<"dati caracterul =";
    cin>>c;
    cout<<" numarul de aparitii :"<<
    aparitii(c,s);
    getch();
}

```

5. Se citește din fișierul MAT . IN o matrice cu n linii și m coloane. Să se afișeze numărul elementelor pătrate perfecte de pe fiecare coloană a matricei utilizând o funcție care să returneze numărul pătratelor perfecte de pe o anumită coloană dată ca parametru.

```

#include<iomanip.h>
#include<conio.h>
#include<fstream.h>
#include<math.h>
int a[30][30],j,n,m;
void citire(int a[30][30],int &n,int &m)
{
    ifstream f;
    int i,j;
    f.open("mat23.in");
    f>>n>>m;
    for(i=1;i<=n;i++)
        for(j=1;j<=m;j++)
            f>>a[i][j];
}
int pp(int c)
{
    int nr=0,i;
    for(i=1;i<=n;i++)

```

```

        if
        (floor(sqrt(a[i][c]))==sqrt(a[i][c]))
nr++;
    return nr;
}
void main()
{
    clrscr();
    citire(a,n,m);
    for(j=1;j<=m;j++)
        cout<<"numarul patratelor perfecte de
pe
        coloana "<<j<<" ="<<pp(j)<<endl;
    getch();
}

```

6. Să se citească și să se afișeze elementele unei matrice cu m linii și n coloane, folosindu-se câte un subprogram pentru fiecare operație.

```

#include<iostream.h>
#include<conio.h>
#include<iomanip.h>
int a[10][10],n,m;
void citire()
{ int i,j;
  for (i=1;i<=m;i++)
    for (j=1;j<=n;j++)
      {
        cout<<"a["<<i<<","<<j<<"]=";
        cin>>a[i][j];
      }
}
void afisare()
{
  int i,j;
  cout<<"Tabloul introdus este:"<<endl;
  for (i=1;i<=m;i++)
    {
      for (j=1;j<=n;j++)
        cout<<setw(4)<<a[i][j];
      cout<<endl;
    }
  cout<<"\n";
}
}

```

```

void main()
{
    clrscr();
    cout<<"n=";cin>>n;
    cout<<"m=";cin>>m;
    citire();
    afisare();
    getch();
}

```

7. Se dă o matrice a cu m linii și n coloane. Să se permute circular spre dreapta cu k poziții toate coloanele matricei. Se va folosi un subprogram care permută circular spre stânga cu o poziție coloanele matricei.

```

#include<iostream.h>
#include<conio.h>
#include<iomanip.h>
int a[10][10],n,m ;

void permut()
{
    int aux[100],i,j;
    for(i=1;i<=m;i++)
        aux[i]=a[i][1];
    for(i=1;i<=m;i++)
        for (j=2;j<=n;j++)
            a[i][j-1]=a[i][j];
    for(i=1;i<=m;i++)
        a[i][n]=aux[i];
}

void main()
{
    clrscr();
    int i,j,k,p;
    cout<<"m=";cin>>m;
    cout<<"n=";cin>>n;
    cout<<"k=";cin>>k;
    p=1;
    for (i=1;i<=m;i++)
        for (j=1;j<=n;j++)
            {
                cout<<"a["<<i<<" , "<<j<<"]=";

```



```

        cin>>a[i][j];
    }
    cout<<"Matricea initiala este: "<<endl;
    for (i=1;i<=m;i++)
    {
        for (j=1;j<=n;j++)
            cout<<setw(4)<<a[i][j];
        cout<<endl;
    }
    cout<<"\n";
    while (p<=k)
    {
        permut();
        p+=1;
    }
    cout<<"Matricea finala este: "<<endl;
    for (i=1;i<=m;i++)
    {
        for (j=1;j<=n;j++)
            cout<<setw(4)<<a[i][j];
        cout<<endl;
    }
    cout<<"\n";
    getch();
}

```

8. Să se scrie un program care inserează între oricare două elemente ale unui vector x de numere reale media aritmetică a vecinilor săi.

```

#include<iostream.h>
#include<iomanip.h>
#include<conio.h>
int n,i,j;
float m;
float x[100];
void calcul (float x[100],int &n)
{
    int i;float m;
    cout<<"n=";cin>>n;
    for (i=1;i<=n;i++)
    {
        cout<<"x["<<i<<"]=";
        cin>>x[i];
    }
}

```

```

    }
    i=1;
    do
    {
        i=i+1;
        m=(x[i-1]+x[i])/2;
        for (j=n;j>=i;j--)
            x[j+1]=x[j];
        x[i]=m;
        n+=1;
        i+=1;
    }
    while (i!=n);
}
void main()
{
    clrscr();
    calcul(x,n);
    for (i=1;i<=n;i++)
        cout<<setw(4)<<x[i]<<" ";
    getch();
}

```

9. Se citește de la tastatură un șir format din n valori. Să se afișeze elementele care apar de cele mai multe ori în cadrul șirului citit.

```

#include<iostream.h>
#include<iomanip.h>
#include<conio.h>
int n,i,j,nr;
long int v[100];
void citire (long int v[20],int &n)
{
    int i;
    cout<<"n=";cin>>n;
    for (i=1;i<=n;i++)
    {
        cout<<"x["<<i<<"]=";
        cin>>v[i];
    }
}
void element(long int v[20])
{
    int i,j,max=0,nr;

```

```

for (i=1;i<=n;i++)
{
nr=0;
for (j=1;j<=n;j++)
if (v[i]==v[j]) nr+=1;
if (max<nr) max=nr;
}
for (i=1;i<=n;i++)
{
nr=1;
for (j=i+1;j<=n;j++)
if (v[i]==v[j]) nr+=1;
if (nr==max) cout<<v[i]<<" ";
}
}
void main()
{
citire(v,n);
element(v);
getch();
}

```

10. Se consideră un șir format din n componente. Să se determine secvența din șir de lungime maximă cu proprietatea că elementele sunt în ordine strict crescătoare.

```

#include<iostream.h>
#include<iomanip.h>
#include<conio.h>
int a[50];
int n,p,pmax,poz,l,lmax,i;
void citire(int a[50],int &n)
{
int i;
cout<<"n=";<<cin>>n;
for (i=1;i<=n;i++)
{
cout<<"a["<<i<<"]="<<endl;
cin>>a[i];
}
}
void lungime(int &l,int &p)
{

```

```

while ((a[p]<a[p+1])&&(p<n))
{
    l+=1;
    p+=1;
}
}
void main()
{
    clrscr();
    citire(a,n);
    p=1;
    do
    {
        l=1;
        poz=p;
        lungime(l,p);
        if (l>lmax)
        {
            lmax=l;
            pmax=poz;
        }
        p+=1;
    }
    while (p<=n);
    cout<<"Secventa maxima: ";
    for (i=pmax;i<=lmax+pmax-1;i++)
        cout<<a[i]<<' ';
    getch();
}

```

Probleme propuse

Următoarele probleme propuse sunt preluate din variantele de bacalaureat 2009.

1. Scrieți în limbajul C/C++ definiția completă a subprogramului **numar**, cu exact doi parametri, care primește prin intermediul parametrului **x** un număr natural nenul de cel mult **2** cifre, și prin intermediul parametrului **y** un număr natural nenul de cel mult **9** cifre. Subprogramul returnează cel mai mare număr natural **z** pentru care există un număr natural **k** astfel încât $z=xk$ și $z \leq y$.

Exemplu: pentru $y=18$ și $x=2$ subprogramul va returna valoarea $16(=24<18)$.

2. Scrieți definiția completă a subprogramului **divizor**, cu trei parametri, prin care primește 3 numere naturale nenule cu cel mult 9 cifre fiecare și returnează numărul divizorilor comuni tuturor celor 3 numere.

Exemplu: dacă numerele primite ca parametri sunt 24, 20 și 12, subprogramul returnează valoarea 3 (divizorii comuni sunt 1, 2 și 4).

3. Subprogramul **Nr** are un singur parametru, **k**, prin intermediul căruia primește un număr natural de cel puțin 3 cifre și cel mult 9 cifre, cu toate cifrele nenule. Subprogramul furnizează tot prin intermediul parametrului **k**, valoarea obținută prin eliminarea primei și ultimei cifre a numărului transmis la apel.

Exemplu: dacă subprogramul primește prin intermediul parametrului **k** valoarea 12438, în urma apelului subprogramului **Nr**, **k** va primi valoarea 243.

Scrieți, în limbajul C/C++, definiția completă a subprogramului **Nr**.

4. Tabloul unidimensional **V**, declarat global, memorează exact 50 de numere întregi: **V1, V2,...,V50**. Subprogramul **Calcul** primește prin intermediul parametrului **k** un număr natural nenul ($k \leq 50$) și furnizează prin intermediul parametrului **S** suma tuturor elementelor pozitive, din tabloul **V**, cu indici mai mari sau egali cu **k** sau 0 dacă toate elementele menționate sunt negative.

a) Scrieți doar antetul subprogramului **Calcul**.

b) Scrieți un program în limbajul C/C++ care citește de la tastatură cele 50 de componente

întregi ale tabloului **V** și două numere naturale nenule **x** și **y** ($x < y \leq 50$). Programul afișează

suma elementelor pozitive din tablou, cu indici cuprinși între **x** și **y** inclusiv, sau 0 dacă toate elementele menționate sunt negative, folosind apeluri utile la subprogramul **Calcul**.

5. Funcția **verif** primește prin intermediul a trei parametri, notați **a, b** și **c**, trei valori naturale nenule, fiecare de maximum patru cifre.

Funcția returnează valoarea **1** dacă cele trei valori pot constitui laturile unui triunghi și valoarea **0** în caz contrar.

a) Scrieți definiția completă a funcției **verif**.

b) Scrieți un program **C/C++** care citește de la tastatură șase valori naturale nenule, fiecare de maximum patru cifre, apoi verifică, utilizând apeluri utile ale funcției **verif**, dacă primele trei numere citite pot constitui laturile unui triunghi și dacă ultimele trei numere citite pot constitui laturile unui triunghi; în caz afirmativ, programul afișează pe ecran mesajul **congruente** dacă cele două triunghiuri sunt congruente sau mesajul **necongruente** dacă cele două triunghiuri nu sunt congruente; dacă cel puțin unul dintre cele două triplete de valori nu pot constitui laturile unui triunghi, programul va afișa pe ecran mesajul **nu**.

6. Funcția **f** primește prin intermediul parametrului **n** un număr natural nenul ($2 \leq n \leq 200$), iar prin intermediul parametrului **a** un tablou unidimensional care conține **n** valori întregi nenule (fiecare dintre aceste valori întregi având cel mult patru cifre).

Funcția returnează valoarea **-1** dacă numărul de valori negative din tabloul **a** este strict mai mare decât numărul de valori pozitive din tablou, valoarea **0** dacă numărul de valori negative din **a** este egal cu numărul de valori pozitive din tablou și valoarea **1** dacă numărul de valori pozitive din tabloul **a** este strict mai mare decât numărul de valori negative din **a**.

Scrieți definiția completă a funcției **f**.

7. Scrieți definiția completă a funcției **f**, care primește prin intermediul parametrului **n** un număr natural nenul ($2 \leq n \leq 200$), iar prin intermediul parametrului **a** un tablou unidimensional care conține **n** valori întregi, fiecare dintre aceste valori întregi având cel mult patru cifre. Funcția returnează valoarea **1** dacă elementele tabloului formează un șir crescător, valoarea **2** dacă elementele tabloului formează un șir descrescător, valoarea **0** dacă elementele tabloului formează un șir constant și valoarea **-1** în rest.

8. **a)** Scrieți definiția completă a subprogramului **dcm**, cu doi parametri, care:

- primește prin parametrii **a** și **b** două valori naturale din intervalul **[1,30000]**

- returnează o valoare naturală reprezentând cel mai mare număr care este atât divizor al lui **a**, cât și divizor al lui **b**.

Exemplu: dacă **a=100** și **b=120**, subprogramul returnează valoarea **20**.

b) Scrieți un program **C/C++** care citește de la tastatură două numere naturale **a** și **b**, numere din intervalul **[1,30000]** și determină, folosind apeluri utile ale subprogramului **dcm**, cel mai mare număr care este divizor al fiecăruia dintre numerele **a** și **b** și are proprietatea că este un produs de două sau mai multe numere prime distincte. Programul afișează pe ecran numărul cu proprietatea cerută, iar dacă nu există un astfel de număr, afișează mesajul **nu exista**.

Exemplu: dacă **a=60** și **b=72**, atunci se afișează **6**
iar dacă **a=100** și **b=75**, atunci se afișează **nu exista**.

9. Subprogramul **ordonare** primește prin parametrul **x** un tablou unidimensional cu cel mult **100** de elemente numere reale, iar prin parametrul **n** un număr întreg ce reprezintă numărul efectiv de elemente ale tabloului **x**. Subprogramul ordonează crescător elementele tabloului și furnizează, tot prin intermediul parametrului **x**, tabloul ordonat.

a) Scrieți numai antetul acestui subprogram.

b) Scrieți un program **C/C++** care citește de la tastatură două numere naturale, **n** și **m** ($1 \leq n \leq 100$ și $m \leq n$), și apoi un șir de **n** numere reale distincte. Folosind apeluri utile ale subprogramului **ordonare**, programul afișează pe prima linie a ecranului, cele mai mari **m** elemente din șirul citit (în ordine crescătoare a valorilor lor), iar pe a doua linie de ecran, cele mai mici **m** elemente din șir (în ordine descrescătoare a valorilor lor). Numerele afișate pe aceeași linie vor fi separate prin câte un spațiu. **(10p.)**

Exemplu : dacă **n=9**, **m=3**, iar șirul este (**14.2**, **60**, **-7.5**, **-22**, **33.8**, **80**, **4**, **10**, **3**) se va afișa pe ecran:

33.8 60 80

3 -7.5 -22

10. Subprogramul **par** primește prin singurul său parametru, **n**, un număr natural nenul cu cel mult **8** cifre și returnează valoarea **1** dacă **n** conține cel puțin o cifră pară, sau returnează valoarea **0** în caz contrar.

Exemplu: pentru **n=723** subprogramul va returna valoarea **1**.

a) Scrieți numai antetul subprogramului **par**.

b) Scrieți un program **C/C++** care citește de la tastatură un număr natural nenul **n** cu cel mult trei cifre, apoi un șir de **n** numere naturale, cu cel puțin două și cel mult **8** cifre fiecare, și afișează pe ecran numărul de valori din șirul citit care au numai cifra unităților pară, celelalte cifre fiind impare. Se vor utiliza apeluri utile ale subprogramului **par**.

Exemplu: dacă **n=4**, iar șirul citit este **7354, 123864, 51731, 570** se va afișa **2** (numerele **7354** și **570** respectă condiția cerută).

11. Funcția **verif** primește prin intermediul parametrului **n** un număr natural format din cel mult 9 cifre, și prin intermediul parametrului **a**, un număr natural nenul ($2 \leq a \leq 9$). Funcția returnează valoarea **1** dacă **n** este un număr format din cifre aparținând intervalului închis $[0, a]$ și valoarea **0** în caz contrar.

a) Scrieți definiția completă a funcției **verif**.

b) Spunem că **n** poate fi o reprezentare în baza **b** ($1 < b \leq 10$), dacă toate cifrele lui **n** sunt strict mai mici decât **b**. Scrieți un program care citește de la tastatură o valoare naturală **n** cu cel mult 9 cifre și, utilizând apeluri ale funcției **verif**, afișează pe ecran, în ordine crescătoare, cu spații între ele, toate valorile lui **b** pentru care valoarea citită **nu** poate fi o reprezentare în baza **b**.

Exemplu: Pentru **n=4101**, se afișează **2 3 4**.

12. Scrieți definiția completă a subprogramului **numar**, cu trei parametri, care primește prin intermediul parametrului **n** un număr natural format din cel mult **9** cifre, iar prin intermediul parametrilor **c1** și **c2** câte o cifră nenulă. Subprogramul caută prima apariție (de la stânga spre dreapta) a cifrei **c1** în **n**, și dacă aceasta apare, o înlocuiește cu **c2**, iar următoarele cifre, dacă există, sunt înlocuite cu câte o cifră **0**. Subprogramul furnizează tot prin **n** numărul astfel obținut. Dacă cifra **c1** nu apare în **n**, atunci valoarea lui **n** rămâne nemodificată.

Exemplu: pentru $n=162448$, $c1=4$ și $c2=7$ valoarea furnizată prin n va fi **162700**.

13. Scrieți definiția completă a subprogramului **numar**, cu patru parametri, care primește prin intermediul parametrului n un număr natural format din cel mult **9** cifre, iar prin intermediul parametrilor $c1$ și $c2$ câte o cifră cu proprietatea $c1 < c2$; subprogramul furnizează prin intermediul celui de al patrulea parametru, x , o valoare obținută prin eliminarea din numărul primit prin parametrul n a fiecărei cifre cuprinse în intervalul închis $[c1, c2]$. Dacă toate cifrele lui n aparțin acestui interval, valoarea furnizată prin x va fi **0**.

Exemplu: pentru $n=162448$, $c1=4$ și $c2=7$, valoarea furnizată prin x va fi **128**.

14. Scrieți definiția completă a subprogramului **reduce**, cu doi parametri, care primește prin intermediul parametrilor a și b două numere naturale formate din cel mult **9** cifre fiecare. Funcția returnează o valoare obținută din numărul a prin însumarea acelor cifre diferite de **0** ale numărului a care **NU** divid numărul b . Dacă nu există asemenea cifre, se va returna valoarea **0**.

Exemplu: pentru $a=184465709$ și $b=18$, cifrele corespunzătoare cerinței sunt **8, 4, 4, 5** și **7**, deci valoarea returnată va fi **28** ($28=8+4+4+5+7$). Dacă $a=2402804$ și $b=8000$, valoarea returnată va fi **0**.

15. Scrieți definiția completă a subprogramului **numar**, cu trei parametri, care primește prin intermediul parametrului n un număr natural format din cel mult **9** cifre, iar prin intermediul parametrilor $c1$ și $c2$ câte o cifră nenulă; subprogramul returnează numărul obținut prin înlocuirea în numărul primit prin parametrul n a fiecărei apariții a cifrei $c1$ cu cifra $c2$. Dacă

$c1$ nu apare în n , subprogramul returnează valoarea n .

Exemplu: pentru $n=12445$, $c1=4$ și $c2=7$ valoarea returnată va fi **12775**.

16. Se consideră subprogramul **CMMDC** care primește prin cei doi parametri, x și y , două numere naturale ($1 \leq x \leq 10000$, $1 \leq y \leq 10000$) și returnează cel mai mare divizor comun al lor.

- a) Scrieți numai antetul subprogramului **CMMDC**.
- b) Fișierul text **NUMERE.IN** conține, pe fiecare linie, câte două numere naturale nenule mai mici sau egale decât **10000**, despărțite printr-un spațiu, reprezentând numitorul și numărătorul câte unei fracții. Scrieți un program **C/C++** care, pentru fiecare linie **k** din fișierul **NUMERE.IN**, citește numitorul și numărătorul fracției de pe această linie și scrie în fișierul text **NUMERE.OUT**, tot pe linia **k**, numitorul și numărătorul acestei fracții, adusă la forma ireductibilă, ca în exemplu. Se vor utiliza apeluri utile ale subprogramului **CMMDC**. (6p.)

Exemplu:

NUMERE . IN	NUMERE . OUT
12 14	6 7
11 12	11 12
2 2	1 1
4 8	1 2

17. Se consideră subprogramul **CMMMC** care primește prin cei doi parametri, **x** și **y**, două numere naturale ($1 \leq x \leq 10000$, $1 \leq y \leq 10000$) și returnează cel mai mic multiplu comun al lor.

- a) Scrieți numai antetul subprogramului **CMMMC**.
- b) Fișierul text **NUMERE.IN** conține, pe fiecare linie, câte două numere naturale nenule mai mici sau egale decât **10000**, despărțite printr-un spațiu. Scrieți un program **C/C++** care, pentru fiecare linie **k** din fișierul **NUMERE.IN**, citește cele două numere de pe această linie și scrie în fișierul text **NUMERE.OUT**, tot pe linia **k**, cel mai mic multiplu comun al acestora, ca în exemplu. Se vor utiliza apeluri utile ale subprogramului **CMMMC**.

Exemplu:

NUMERE . IN	NUMERE . OUT
12 14	84
11 12	132
4 8	8

18. Scrieți în **C/C++** definiția completă a subprogramului **suma** care are doi parametri:

- **n**, prin care primește un număr natural ($1 \leq n \leq 100$);

- **v**, prin care primește un tablou unidimensional cu **n** elemente, numere întregi, fiecare având exact trei cifre.

Funcția returnează suma elementelor din tablou care au prima cifră egală cu ultima cifră.

19. Scrieți în **C/C++** definiția completă a subprogramului **suma** care are doi parametri:

- **n**, prin care primește un număr natural ($1 \leq n \leq 100$);

- **v**, prin care primește un tablou unidimensional cu **n** elemente, numere întregi situate în intervalul $[10, 30000]$. Funcția returnează suma numerelor din tabloul **v** care au ultimele două cifre identice.

Exemplu: dacă $n=4$ și $v=(123, 122, 423, 555)$ funcția va returna **677** ($=122+555$).

20. Scrieți în **C/C++** definiția completă a subprogramului **medie** care are doi parametri:

- **n**, prin care primește un număr natural ($1 \leq n \leq 100$);

- **v**, prin care primește un tablou unidimensional cu **n** elemente, numere naturale, fiecare element având cel mult patru cifre.

Subprogramul returnează media aritmetică a elementelor din tablou.

21. **a)** Scrieți definiția completă a unui subprogram **divi** cu doi parametri, care primește prin intermediul parametrului **n** un număr natural nenul cu cel mult **6** cifre și returnează prin intermediul parametrului **d** cel mai mic divizor propriu al lui **n** sau **0** în cazul în care **n** nu are niciun divizor propriu. De exemplu, pentru $n=15$, **d** va avea valoarea **3**.

b) Scrieți programul **C/C++** care citește de la tastatură un număr natural nenul **n** ($1 < n < 1000000$) și afișează pe ecran cel mai mare număr natural mai mic decât **n** care îl divide pe **n**. Se vor folosi apeluri utile ale subprogramului **divi**.

Exemple: dacă $n=120$, se afișează pe ecran **60**, iar dacă $n=43$, se afișează **1**.

22. Scrieți definiția completă a subprogram **max**, cu trei parametri, **a**, **b**, **c**, care primește prin intermediul parametrilor **a** și **b** două numere reale cu exact două cifre la partea întregă și exact două zecimale fiecare. Subprogramul determină cel mai mare număr real dintre

următoarele patru valori: **a**, **b** și numerele reale obținute din **a** și **b** prin interschimbarea părții întregi cu partea fracționară în cadrul aceluiași număr. Această valoare este furnizată prin intermediul parametrului real **c**.

Exemplu: dacă **a=33.17** și **b=15.40**, **c** va avea valoarea **40.15** (cea mai mare valoare dintre **33.17**, **15.40**, **17.33** și **40.15**)

23. **a)** Scrieți doar antetul unui subprogram **prim** cu doi parametri, care primește prin intermediul parametrului **n** un număr natural cu cel mult patru cifre și returnează prin intermediul parametrului **p** valoarea **1** dacă **n** este prim și **0** în caz contrar.

b) Scrieți un program **C/C++** care citește de la tastatură un număr natural **n** ($3 < n < 10000$) și afișează pe ecran, despărțite prin câte un spațiu, primele **n** numerele prime, folosind apeluri utile ale subprogramului **prim**.

Exemplu: pentru **n=4** pe ecran vor fi afișate numerele **2 3 5 7**

24. Scrieți definiția completă a unui subprogram **s** cu trei parametri care primește prin intermediul parametrului **n** un număr natural de maximum **9** cifre, prin intermediul parametrului **c** o cifră și furnizează prin intermediul parametrului **k** numărul de cifre ale numărului **n** care aparțin intervalului **[c-1,c+1]**.

Exemplu: pentru **n=1233** și **c=3**, **k** va avea valoarea **3**, iar pentru **n=650** și **c=3**, **k** va avea valoarea **0**.

25. **a)** Scrieți definiția completă a funcției **UltimaCifra** care primește prin cei doi parametri **a** și **b** câte un număr natural ($0 < a < 1000000$, $0 < b < 1000000$), calculează în mod eficient din punct de vedere al timpului de executare și returnează ultima cifră a numărului **a^b** (a la puterea b).

b) Fișierul text **SIR.IN** conține pe prima sa linie un număr natural **n** ($0 < n < 1001$), iar pe fiecare dintre următoarele **n** linii câte o pereche de numere naturale, **x_i**, **y_i** ($1 \leq i \leq n$, $x_i \leq 30000$, $y_i \leq 30000$).

Scrieți programul **C/C++** care citește numerele din fișierul **SIR.IN** și scrie în fișierul text **SIR.OUT** ultima cifră expresiei:

$$\sum_{i=1}^n x_i^{y_i}, \text{ folosind apeluri ale funcției } \text{UltimaCifra}.$$

26. a) Scrieți definiția completă a subprogramului **Cifra**, cu doi parametri, **n** și **x**, care primește prin intermediul parametrului **n** un număr natural de cel mult nouă cifre și furnizează prin parametrul **x** cea mai mare cifră a numărului transmis prin parametrul **n**.

b) Scrieți un program **C/C++** care citește de la tastatură un număr natural **n**, $n < 100000000$, și afișează pe ecran mesajul **Da** în cazul în care numărul citit este format

doar din cifre aparținând mulțimii **{0,1,2,3,4}** și afișează mesajul **Nu** în caz contrar.

27. a) Scrieți definiția completă a subprogramului **Cautare**, cu trei parametri, **n**, **x** și **v**, care primește prin parametrul **n** un număr natural ($1 \leq n \leq 1000$), prin parametrul **x** un tablou unidimensional format din **n** componente (numere întregi de cel mult patru cifre fiecare: **x1**, **x2**, ..., **xn**) memorate în ordine crescătoare și prin parametrul **v** un număr întreg de cel mult patru cifre, diferit de oricare dintre elementele tabloului unidimensional **x**. Subprogramul va căuta, în mod eficient din punct de vedere al timpului de executare, poziția pe care ar trebui inserată valoarea **v** în șirul **x** astfel încât să se obțină tot un șir ordonat și returnează această poziție.

b) Fișierul text **sir.in** conține cel mult **1000** numere naturale de maximum patru cifre fiecare, numerele fiind diferite două câte două și despărțite prin câte un spațiu.

Scrieți un program **C/C++** care citește numerele din fișierul **sir.in** și, folosind apeluri utile ale subprogramului **Cautare**, construiește în memorie un tablou unidimensional care va conține toate numerele din fișierul **sir.in** ordonate crescător. Programul scrie în fișierul text **sir.out** șirul obținut, câte **10** elemente pe un rând, elementele de pe același rând fiind despărțite printr-un singur spațiu.

Exemplu: dacă fișierul **sir.in** conține numerele: **7 -5 635 -456 0 8 587 -98 65 3 -8**

atunci după executarea programului fișierul **sir.out** va conține:

-456 -98 -8 -5 0 3 7 8 65 587

635

28. a) Scrieți definiția completă a subprogramului **Ecuatie** care primește prin parametrii **a**, **b** și **c** trei numere întregi, $a \neq 0$, de cel

mult patru cifre fiecare, reprezentând coeficienții ecuației de gradul al II-lea: $ax^2 + bx + c = 0$. În funcție de soluțiile ecuației subprogramul va returna:

- cea mai mare dintre soluții dacă ecuația are două soluții reale distincte, dintre care cel puțin una pozitivă.
- una dintre soluții dacă ecuația are două soluții egale și pozitive.
- **-32000** în celelalte cazuri.

b) Se consideră șirul s : **1, 1, 2, 1, 2, 3, 1, 2, 3, 4, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 6, 1, 2, ...**

Pentru un număr natural k , $0 < k \leq 10000$, se cere să se determine valoarea elementului ce se află pe poziția k în șirul s .

Exemplu: pentru $k=5$ numărul cerut este **2**.

Scrieți un program C/C++ care citește de la tastatură valoarea numărului natural k și, prin apeluri utile ale funcției **Ecuatie**, determină valoarea elementului ce se află pe poziția k în șirul s , folosind un algoritm eficient din punctul de vedere al spațiului de memorie alocat și al timpului de executare. Valoarea astfel determinată se va scrie în fișierul text **sir.out**.

29. a) Scrieți definiția completă a subprogramului **Palindrom** care primește prin intermediul parametrului n un număr natural de cel mult nouă cifre și returnează **1** dacă acesta este palindrom și **0** în caz contrar.

b) Fișierul text **NUMERE.IN** conține cel mult **10000** numere naturale de cel mult nouă cifre fiecare, numerele fiind despărțite prin câte un spațiu. Cel puțin unul dintre numere este palindrom.

Scrieți programul C/C++ care citește numerele din fișierul **NUMERE.IN** și, folosind apeluri utile ale subprogramului **Palindrom** determină în mod eficient, din punct de vedere al memoriei utilizate și al timpului de executare, care este cel mai mare număr palindrom citit și de câte ori apare el în fișierul **NUMERE.IN**. Programul scrie în fișierul text **NUMERE.OUT** numărul astfel determinat precum și numărul de apariții ale acestuia, pe rânduri diferite.

Exemplu:

NUMERE.IN	NUMERE .OUT
23 565 78687 7887 7865 78687	78687
7887 23 78687 98798	3

30. a) Să se scrie definiția completă a subprogramului **calcul**, care primește prin intermediul celor doi parametri ai săi două numere întregi, **n** și **k** ($1 \leq n \leq 100000000$ și $1 \leq k \leq 9$), și returnează cifra de rang **k** a numărului **n**. Rangul unei cifre este numărul său de ordine, numerotând cifrele de la dreapta la stânga; cifra unităților având rangul **1**. Dacă numărul **k** este mai mare decât numărul de cifre ale lui **n**, atunci funcția returnează valoarea **-1**.

Exemplu: dacă **n=9243** și **k=3**, în urma apelului se va returna **2**.

b) Scrieți un program **C/C++** care citește de la tastatură un număr natural **n** cu cel mult **8** cifre. Programul va verifica, utilizând apeluri ale subprogramului **calcul**, dacă orice cifră a lui **n** are rangul cifrei mai mare sau egal cu valoarea cifrei respective și va afișa mesajul **Da** în caz afirmativ și mesajul **Nu** în caz contrar.

Exemplu : pentru **n=4160** se va afișa **Nu**.

rang **4 3 2 1**

cifră **4 1 6 0**

CAPITOLUL IV STRUCTURI DE DATE ALOCATE STATIC

1. Creați o structură de tip listă în care elementele sunt numere întregi. Să se realizeze printr-un meniu principalele operații pe listă:

- a) crearea unei liste cu un număr dat de elemente;
- b) adăugarea unui element pe prima poziție;
- c) adăugarea unui element pe ultima poziție;
- d) ștergerea elementului de pe prima poziție;
- e) ștergerea elementului de pe ultima poziție;
- f) afișarea elementelor dintr-o listă.

```
#include<iostream.h>
#include<conio.h>
typedef int Lista[200];
Lista l;
int n;
void init(int &n)
{ n=0;}
void creare(Lista l, int &n)
{
    cout<<"Dati numarul de elemente al listei=";
    cin>>n;
    for(int i=1;i<=n;i++)
    {
        cout<<"Elementul "<<i<<" = ";
        cin>>l[i];
    }
}
void addInc(Lista l, int &n)
{
    int newEl;
    cout<<"Noul element : "; cin>>newEl;
    for(int i=n;i>=1;i--) l[i+1]=l[i];
    l[1]=newEl;
    n++;
}
void addSf(Lista l,int &n)
{
    int newEl;
    cout<<"Noul element : "; cin>>newEl;
```



```

    n++;
    l[n]=newEl;
}
void remInc(Lista l, int &n)
{
    if (n==0)
        cout<<"Lista este goala!"<<endl;
    else
        {
            for (int i=2;i<=n;i++) l[i-1]=l[i];
            n--;
            cout<<"Primul element a fost eliminat!";
            cout<<endl;
        }
}
void remSf(Lista l, int &n)
{
    if (n==0) cout<<"Lista este goala!"<<endl;
    else n--;
}
void afisLista(Lista l, int n)
{
    if (n==0) cout<<"lista nu are elemente! "<<endl;
    else {
        for(int i=1;i<=n ;i++)
            cout<<l[i]<<" ";cout<<endl;
    }
}
void main(void)
{
    int id=0;
    init(n);
    do
    {
        clrscr();
        cout<<"1. Creare lista: "<<endl;
        cout<<"2. Adaugare la inceput: "<<endl;
        cout<<"3. Adaugare la sfarsit: "<<endl;
        cout<<"4. Eliminare de la inceput: "<<endl;
        cout<<"5. Eliminare de la sfarsit: "<<endl;
        cout<<"6. Afisare: "<<endl;
        cout<<"0. Exit "<<endl;
    }
}

```

```

cout<<"Introduceti identificatorul meniului:";
cin>>id;
switch (id)
{
    case 1 : creare(L,n); break;
    case 2 : addInc(L,n); break;
    case 3 : addSf(L,n) ; break;
    case 4 : remInc(L,n); break;
    case 5 : remSf(L,n) ; break;
    case 6 : afisLista(L,n); break;
    case 7 : addPozK(L,n); break;
    case 8 : remPozK(L,n); break;
    case 9 : addXafterY(L,n); break;
    case 10: remVal(L,n); break;
    case 11: modValPozK(L,n); break;
    case 12: modVal(L,n); break;
    case 13: remDup(L,n); break;
} cout<<" Apasa o tasta pentru a
continua!";
    getch();
} while(id);
}

```

2. Creați o structură de tip stivă în care elementele sunt numere întregi. Să se realizeze printr-un meniu principalele operații pe stivă:

- a) crearea unei stive cu număr dat de elemente;
- b) adăugarea unui element în stivă;
- c) ștergerea unui element din stivă;
- d) afișarea elementelor din stivă;
- e) afișarea elementului din vârful stivei.

```

#include<iostream.h>
#include<conio.h>
typedef int Stiva[200];
Stiva S;
int n;
void init(int &n)
{
    n=0;
}
void creare(Stiva s, int &n)
{

```

```

    cout<<"Dati numarul de elemente al stivei = ";
    cin>>n;
    for(int i=1;i<=n;i++)
    {
        cout<<"Elementul "<<i<<" = ";
        cin>>s[i];
    }
}
void add(Stiva s, int &n)
{
    int newEl;
    cout<<"Noul element: "; cin>>newEl;
    n++;
    s[n]=newEl;
}
void rem(Stiva s, int &n)
{
    if (n==0) cout<<"Stiva este goala!"<<endl;
        else n--;
}
void afis(Stiva s, int n)
{
    if(n==0) cout<<"Stiva nu are elemente"<<endl;
        else {
            for(int i =1;i<=n;i++) cout<<s[i]<<" ";
            cout<< endl;
        }
}
void element(Stiva s, int n)
{
    if(n==0) cout<<"Stiva nu are elemente!"<<endl;
        else cout<<"Elementul din varf este: "<<s[n];
    cout<<endl;
}
void main(void)
{
    int id=0;
    init(n);
    do
    {
        clrscr();
        cout<<"1. Creare Stiva: "<<endl;

```

```

cout<<"2. Adaugare: "<<endl;
cout<<"3. Eliminare: "<<endl;
cout<<"4. Afisare: "<<endl;
cout<<"5. Afisare element din varful
      stivei: "<<endl;
cout<<"0. Exit "<<endl;
cout<<"Introduceti identificatorul meniului:";
cin>>id;
switch (id)
{
    case 1: creare(S,n); break;
    case 2: add(S,n); break;
    case 3: rem(S,n); break;
    case 4: afis(S,n); break;
    case 5: element(S,n); break;
}
cout<<"Apasa o tasta pentru a continua!";
} while(id);
}

```

3. Operațiile ce se pot efectua asupra unei cozi sunt:

- a)** crearea unei cozi cu număr dat de elemente;
- b)** adăugarea unui element în coadă;
- c)** ștergerea unui element din coadă;
- d)** afișarea elementelor din coadă.

```

#include<iostream.h>
#include<conio.h>
typedef int Coadă[200];
Coadă C;
int n;
void init(int &n)
{ n=0;}
void creare(Coadă c, int &n)
{
    cout<<"Dati numarul de elemente ="; cin>>n;
    for(int i=1;i<=n;i++)
    {
        cout<<"Elementul "<<i<<" = ";
        cin>>c[i];
    }
}

```

```

}
void add(Coada c,int &n)
{
    int newEl;
    cout<<"Noul element: "; cin>>newEl;
    n++;
    c[n]=newEl;
}
void rem(Coada c, int &n)
{
    if (n==0 ) cout<<"Coadă este goală!"<<endl;
        else {
            for(int i=1;i<n;i++)
                c[i]=c[i+1];
            n--;
        }
}
void afis(Coada c, int n)
{
    if (n==0) cout<<"Coadă nu are elemente!"<<endl;
        else {
            for(int i =1;i<=n;i++)
                cout<<c[i]<<" ";cout<<endl;
        }
}
void main(void)
{
    int id=0;
    init(n);
    do
    {
        clrscr();
        cout<<"1. Creare: "<<endl;
        cout<<"2. Adaugare: "<<endl;
        cout<<"3. Eliminare: "<<endl;
        cout<<"4. Afisare: "<<endl;
        cout<<"0. Exit "<<endl;
        cout<<"Introduceti identificatorul meniului:";
        cin>>id;
        switch (id)
        {
            case 1: creare(C,n); break;

```

```

        case 2: add(C,n) ; break;
        case 3: rem(C,n) ; break;
        case 4: afis(C,n) ; break;
    }
    cout<<"Apasa o tasta pentru a continua! ";
    getch();
} while(id);
}

```

Probleme propuse

Următoarele probleme propuse sunt preluate din variantele de bacalaureat 2009.

1. Într-o stivă au fost introduse, în această ordine, valorile **10, 5, 4**. Dacă se notează cu **PUSH(x)** operația prin care se adaugă valoarea **x** în vârful stivei, și cu **POP** operația prin care se extrage elementul din vârful stivei, care este conținutul stivei după executarea următoarelor operații? **POP; PUSH(7); POP; POP; PUSH(9)**;

2. Alegeți: Stiva este o structură de date care poate fi descrisă astfel:

a. oricare element poate fi extras

b. ultimul element introdus în stivă este ultimul care poate fi extras

c. primul element introdus în stivă este primul care poate fi extras

d. primul element introdus în stivă este ultimul care poate fi extras

3. Într-o stivă au fost introduse în această ordine, numerele **5, 7, 3, 8**. Precizați numărul minim de elemente care trebuie extrase din stivă pentru a fi siguri că s-a extras inclusiv elementul cu valoarea **3** și care este elementul aflat în vârful stivei după extragerea acestui element?

4. Într-o stivă au fost introduse, în această ordine, numerele **5, 7, 3, 8**. Scrieți care este numărul minim de elemente care trebuie extrase din stivă pentru a fi siguri că s-a extras inclusiv elementul cu valoarea **7** și care este numărul de elemente rămase în stivă după extragerea acestui element.

5. Care va fi valoarea elementului aflat în vârful unei stive inițial vidă și care este numărul de elemente rămase în stivă, după

efectuarea, în această ordine, a următoarelor operații: se introduce valoarea **3**; se introduce valoarea **7**; se introduce valoarea **5**; se extrage un element; se introduce valoarea **2**; se introduce valoarea **4**; se extrage un element.

6. Care vor fi valorile primului și ultimului element extras dintr-o **coadă** inițial vidă, dacă se efectuează următoarele operații, în această ordine: se introduce valoarea **5**; se introduce valoarea **4**; se extrage un element; se introduce valoarea **2**; se introduce valoarea **7**; se extrage un element.

7. Stiva **S** și coada **C** memorează numere întregi. În ambele se introduc, în ordine, numerele **1, 2, 3, 4**. Se notează cu $S \Rightarrow C$ operația de extragere a unui element din stiva **S** și adăugarea acestuia în coada **C**, iar cu $C \Rightarrow S$ operația de eliminare a unui element din coada **C** și introducerea acestuia în stiva **S**. După executarea următoarei secvențe de operații: $C \Rightarrow S; C \Rightarrow S; S \Rightarrow C; C \Rightarrow S; C \Rightarrow S$;

a) care este ultima valoare introdusă în stiva **S**?

b) care este ultima valoare care a fost adăugată în coada **C**?

8. Într-o stivă ce memorează numere întregi se introduc, în ordine, următoarele numere: **1,2,3,4,5,6,7**. Câte numere trebuie să eliminăm din stivă astfel ca în vârful stivei să se găsească numărul **5**?

9. Se consideră o coadă în care inițial au fost introduse, în această ordine, elementele **1,2,3,4,5,6**. Dacă se notează cu $AD(x)$ operația prin care se adaugă un element cu informația **x** în coadă și cu $EL()$ operația prin care se elimină un element din coadă, care este elementul aflat în mijlocul cozii și care este suma elementelor aflate în coadă după executarea secvenței de operații: $EL(); AD(7); AD(8); EL(); EL()$;

10. Se consideră o coadă în care inițial au fost introduse, în această ordine, elementele **1,2,3,4,5,6,7,8,9,10**. Dacă se notează cu $AD(x)$ operația prin care se adaugă un element cu informația **x** în coadă și cu $EL()$ operația prin care se elimină un element din coadă, care este valoarea memorată în primul element al cozii după executarea secvenței de operații: $EL(); EL(); AD(1); AD(2); EL(); EL(); ?$

CAPITOLUL V RECURSIVITATE

Itemii următori sunt preluați din variantele de bacalaureat 2009.

1. Se consideră subprogramul **f** definit alăturat. Ce valoare are **f(250)**?

```
int f(int x)
{
    if(x%3==0) return 0;
    else return 1+f(x/3);
}
```

2. Se consideră subprogramul **f** definit alăturat. Ce valoare are **f(5)**?
Dar **f(100)**?

```
long f(int n)
{
    if(n<0) return 0;
    else return f(n-2)+n;
}
```

3. Funcția **f** are definiția alăturată. Dacă **f(x)** are valoarea **10100**, care este valoarea lui **x**?

```
long f(int n)
{
    if(n<=0) return 0;
    else return f(n-1)+2*n;
}
```

4. Subprogramul **f** are definiția alăturată. Ce valoare are **f(4)**? Dar **f(11)**?

```
int f(int x)
{if(x<1) return 1;
else return f(x-3)+1;
}
```

5. Subprogramul **f** are definiția alăturată. Ce valoare are **f(6,5)**? Dar **f(5,10)**?

```
int f(int x,int y)
```



```

    {if(x==y) return x;
    else if(x<y) return f(x+1,y-1);
    else return f(x-1,y);
    }

```

6. Subprogramul **f** are definiția alăturată. Ce valoare are **f(3)**? Dar **f(10)**?

```

int f(int x)
{if(x==0) return 0;
else return f(x-1)+2;}

```

7. Subprogramul **f** are definiția alăturată. Ce valoare are **f(7)**? Dar **f(100)**?

```

int f(int x)
{if(x%6==0) return x;
else return f(x-1);
}

```

8. Pentru definiția alăturată a subprogramului **f**, ce valoare are **f(3)**? Dar **f(8)**?

```

int f(int x)
{
if(x<=4) return x*x-3;
return f(x-3)+4;
}

```

9. Pentru definiția alăturată a subprogramului **sc**, stabiliți ce valoare are **sc(10)**. Dar **sc(901324)**?

```

int sc(long x)
{
if(x<10) return x;
return sc(x/10)+x%10;
}

```

10. Considerăm subprogramul **f** definit alăturat. Ce valoare are **f(7,11)**? Dar **f(11,7)**?

```

int f(int x, int y)
{
if(x<=y) return x-y;
}

```

```
return f(y-x, x-1)+3;
}
```

11. Subprogramul **scif** returnează suma cifrelor unui număr natural transmis ca parametru.

Care este valoarea expresiei **scif(scif(518)+scif(518))**?

12. Se consideră definit subprogramul **f**.

Scrieți două valori naturale, **x1** și **x2** (**x1**≠**x2**, **x1**<**12** și **x2**<**12**) pentru care **f(x1)=f(x2)**.

```
int f(int i)
{
if (i>12) return 1;
else return 1+f(i+2);
}
```

13. Subprogramul **re** este definit alăturat. Ce valoarea are **re(1)**? Dar **re(14)**?

```
int re(int i)
{
if (i<9) return 3+re(i+2);
else
if (i==9) return -2;
else return 1+re(i-1);
}
```

14. Subprogramul **f** este definit alăturat.

Ce valoare are **f(8,4)**?

```
int f (int x, int y)
{
if (x<y) return 1+f(x+1, y);
if (y<x) return 1+f(y+1, x);
return 1;
}
```

15. Funcția **f** are definiția alăturată:

a) Ce valoare are **f(16)**?

b) Scrieți cea mai mare valoare de două cifre pe care o poate avea n astfel încât $f(n)$ să fie egal cu 2.

```
int f(int n)
{ if (n<=0) return -1;
  if (n%2==0) return 0;
  if (n%3==0) return 0;
  return 1+f(n-10);
}
```

16. Funcția f are definiția alăturată.

a) Ce valoare are $f(17)$?

b) Ce valoare are $f(22)$?

```
int f(int n)
{if (n<=9) return 0;
 if (n%4==0) return 0;
 return 1+f(n-3);
}
```

17. Pentru subprogramul **suma** definit alăturat, scrieți valoarea expresiei **suma(5,4)**.

```
int suma (int a,int b)
{ if (a==0 && b==0) return 0;
  else if (a==0) return 1+suma(a,b-1);
  else return 1+suma(a-1,b);
}
```

18. Se consideră subprogramul, f , definit alăturat.

a. Ce valoare are $f(100)$?

b. Scrieți o valoare pentru x astfel încât $f(x)=1$

```
int f(int n)
{ if(n==0) return 0;
  else return n%2+f(n/2);
}
```

19. Funcția F are definiția alăturată. Ce valoare are $F(5)$?

```
int F(int x)
{if(x!=0) return x+F(x-1);
 else
```

```
return x;
}
```

20. Funcția **F** are definiția alăturată. Ce valoare are **F(18)**?

```
int F(int x){
if (x<=1) return x;
else return x+F(x-2);
}
```

21. Funcția **F** are definiția alăturată. Ce valoare are **F(3)**?

```
int F(int n)
{if(n==0 || n==1) return 1;
else
return 2*F(n-1)+2*F(n-2);}
```

22. Pentru funcțiile **f** și **g** definite mai jos, scrieți care este rezultatul returnat la apelul **g(11)**.

Dar rezultatul returnat la apelul **f(6)**?

```
long g(long x)
{ if (x>9)
return (x/10 + x%10);
else
return x;
}
```

```
long f(int c)
{ if (c<1)
return 1;
else
return g(c+f(c-1));
}
```

23. Pentru funcțiile **f1** și **f2** definite alăturat, stabiliți care este valoarea lui **f1(3)**. Dar **f2(41382)**?

```
long f1(int c)
{ if (c%2==1) return 1;
else return 2;
```

```

}
long f2(long n)
{ if (n==0) return 0;
else return f1(n%10)+f2(n/10);
}

```

24. Pentru funcția **f** definită alăturat, stabiliți care este valoarea **f(5)**.
Dar **f(23159)**?

```

int f(int n){
int c;
if (n==0) return 9;
else
{c=f(n/10);
if (n%10<c) return n%10;
else return c;
}
}

```

25. Se consideră subprogramul cu definiția alăturată. Ce valoare are **f(3,1)**?

```

int f(int n,int y)
{ if(n!=0)
{ y=y+1;
return y+f(n-1,y);
}
else return 0;
}

```

26. Se consideră subprogramul **f** cu definiția alăturată. Ce valoare are **f(1213111,1)**?

```

int f (long n, int k){
if (n!=0)
if(n%10==k)
return 1+f(n/10,k);
else return 0;
else return 0;}

```

27. Se consideră subprogramul alăturat:

```
int f(int a, int b)
{if (b<1) return -1;
else
if (a%b==0)
return 1+f(a/b,b);
else
return 0; }
```

Ce valoare are **f(15,2)**? Dar **f(128,2)**?

28. Pentru definiția de mai jos a subprogramului **f**, ce se afișează ca urmare a apelului **f(121,1)** ;?

```
void f(long n, int i)
{ if(n!=0)
if(n%3>0)
{ cout<<i; f(n/3,i+1); }
}
```

29. Pentru definiția de mai jos a subprogramului **f**, ce se afișează ca urmare a apelului **f(12345)** ;?

```
void f(long n)
{ cout<<n%10;
if(n!=0)
{ f(n/100); cout<<n%10; }
}
```

30. Pentru definiția alăturată a subprogramului **f**, ce se afișează ca urmare a apelului **f(26)** ;?

```
void f (int x)
{if(x>0)
if(x%4==0)
{ cout<<'x' ;
f(x-1); }
else
{ f(x/3);
cout<<'y' ; }}
```

31. Pentru definiția alăturată a subprogramului **f**, ce se afișează ca urmare a apelului **f(15, 2)** ;?

```
void f (int n, int x)
{ if(x>n)
cout<<0;
else
if(x%4<=1) f(n, x+1);
else
{ f(n, x+3);
cout<<1;
}
}
```

32. Pentru subprogramul **f** definit mai jos, ce se afișează ca urmare a apelului **f(3,17)**?

```
void f ( int a, int b)
{ if(a<=b)
{ f(a+1,b-2); cout<<'*';}
else cout<<b;
}
```

33. Se consideră subprogramul **f** definit alăturat. Ce se va afișa în urma apelului **f(12345)** ;?

```
void f(long int n)
{ if (n!=0)
{if (n%2 == 0)
cout<<n%10;
f(n/10);
}
}
```

34. Se consideră subprogramul **f**, descris alăturat. Ce se va afișa în urma apelului **f(3)** ;?

```
void f(int n)
{ if (n!=0)
{ if (n%2==0)
cout<<n<<' ';
f(n-1);
```

```

cout<<n<<' ';
}
else cout<<endl;
}

```

35. Subprogramul **f** are definiția alăturată. Ce se va afișa în urma apelului **f(12345)**?

```

void f(long n)
{if (n>9)
{cout<<n/100;
f(n/10);
}
}

```

36. Funcția **f** are definiția alăturată. Ce se va afișa în urma apelului **f(12345, 0)** ?

```

void f(long n, int i)
{if (i<n%10)
{cout<<n%10;
f(n/10,i+1);
}
}

```

37. Considerăm subprogramul recursiv definit alăturat. Ce se va afișa în urma apelului de mai jos?
f('C') ;

```

void f(char c)
{ if (c>'A') f(c-1);
cout<<c;
if (c>'A') f(c-1);}

```

38. Subprogramul **afis** este definit alăturat. Ce se afișează ca urmare a apelului **afis(8)** ; ?

```

void afis (int n)
{
cout<<n;
for (int i=n/2;i>=1;i--)
if (n%i==0)afis (i);}

```


39. Subprogramul **afis** este definit alăturat. Ce se afișează ca urmare a apelului **afis(8)**; ?

```
void afis (int n)
{
    cout<<n;
    for (int i=n/2;i>=1;i--)
        if(n%i==0) afis(i);
}
```

40. Subprogramul **scrie** este definit alăturat. Ce se afișează ca urmare a apelului **scrie(2,6)**; ?

```
void scrie (int x,int y)
{
    cout<<x<<y;
    if(x<y)
    {
        scrie(x+1,y-1);
        cout<<(x+y)/2;
    }
}
```

41. Ce se afișează ca urmare a apelului **p(123)**; dacă subprogramul **p** are definiția alăturată?

```
void p (int x)
{cout<<x;
  if(x!=0){p(x/10);
  cout<<x%10; }}
```

42. Subprogramul **f** este definit alăturat. Ce se afișează ca urmare a apelului **f(1,3)**;?

```
void f (int x,int y)
{int i;
  for (i=x;i<=y;i++)
  {
    cout<<i;
    f(i+1,y);
  }
}
```

43. Se consideră subprogramul recursiv **f1** definit alăturat. Ce se va afișa în urma apelului **f1(5)**?

```
void f1(int x)
{ if (x<=9)
{ cout<<x+1;
f1(x+2);
cout<<x+3; }
}
```

44. Subprogramul **afis** este definit alăturat. Ce se va afișa în urma apelului **afis(17)**?

```
void afis(int x)
{ if (x>3)
{ cout<<x-1;
afis(x/3);
cout<<x+1;
}
}
```

45. Știind că **p** este un vector cu **3** componente întregi (vector declarat global), stabiliți cu ce trebuie înlocuite alfa α și beta β în definiția subprogramului **G** alăturat astfel încât în urma apelului **G(0)** să se afișeze toate numerele de **3** cifre nenule, fiecare număr fiind afișat o singură dată.

```
void G(int k)
{ int i;
for(i=1;i<=alfa  $\alpha$ ;i++)
{ p[k]=i;
if(beta  $\beta$ )G(k+1);
else
cout<<p[0]<<p[1]<<p[2]<<endl;
} }
```

Probleme rezolvate

1. Se citesc de la tastatură două numere naturale nenule a și b . Să se calculeze printr-o funcție recursivă cel mai mare divizor comun al numerelor utilizând algoritmul lui Euclid.

```

#include<conio.h>
#include<iostream.h>
int a,b;
int cmmdc(int a,int b)
{
    if (b==0) return a;
    else return cmmdc(b,a%b);
}
void main()
{
    cout<<"a="; cin>>a;
    cout<<"b="; cin>>b;
    cout<<"cmmdc="<<cmmdc(a,b);
    getch();
}

```

2. Se dă un număr natural n citit de la tastatură. Să se verifice printr-o funcție recursivă dacă numărul este prim sau nu.

```

#include<iostream.h>
#include<conio.h>
int n;
int prim(int n,int d)
{
    if((n==0)|| (n==1)) return 0;
    else
        if(d<=n/2)
            if (n%d==0) return 0;
            else return prim(n,d+1);
            else return 1;
}
void main()
{cout<<"dati n =";cin>>n;
  if (prim(n,2)) cout<<"Numarul este prim";
  else cout<<"Numarul nu este prim";
  getch();
}

```

3. Se citește de la tastatură un număr natural de cel mult 9 cifre. Să se calculeze suma cifrelor numărului printr-o funcție recursivă.

```

#include<iostream.h>
#include<conio.h>
long int n;
int suma_cifre(long int n)

```

```

{
    if (n==0) return 0;
    else
        return n%10+suma_cifre(n/10);
}

void main()
{
    cout<<"\n"<<"Dati n =";
    cin>>n;
    cout<<"Suma cifrelor  =
"<<suma_cifre(n);
    getch();
}

```

4. Se citește de la tastatură un șir de caractere având lungimea maximă de 30. Să se realizeze o funcție recursivă care să returneze numărul de vocale.

```

#include<conio.h>
#include<iostream.h>
#include<string.h>
char *voc="aeiouAEIOU", s[30];
int nr_voc(char s[30],char *p)
{
    if (!(strpbrk(s,p))) return 0;
    else
        { s=strpbrk(s,p);
          s++;
          return 1+nr_voc(s,p);
        }
}

void main()
{
    char *p;
    cout<<"Dati sirul : ";cin>>s;
    cout<<"Numarul de vocale="
"<<nr_voc(s, voc);
    getch();
}

```

5. Să se citească de la tastatură un șir format din n componente, utilizând o funcție recursivă.

```

#include<iostream.h>
#include<conio.h>
int a[20],n;
void citire(int a[20],int i)
{
    if (i<=n)
        {
            cout<<"a["<<i<<"]="; cin>>a[i];
            citire(a,i+1);
        }
}
void afisare()
{ int i;
  for(i=1;i<=n;i++)
    cout<<a[i]<<" ";
  cout<<endl;
}
void main()
{
    cout<<"Dati numarul de elemente = ";
    cin>>n;
    citire(a,1);
    afisare();
    getch();
}

```

6. Din fişierul text *MAT.IN* se citeşte o matrice pătratică de dimensiune $n \times n$. Să se afişeze, printr-o funcţie recursivă, elementele de sub diagonală principală, exclusiv diagonală.

```

#include<iostream.h>
#include<conio.h>
#include<fstream.h>
#include<iomanip.h>
int a[20][20],n;
void citire(int a[20][20],int &n)
{
    ifstream f;
    int i,j;
    f.open("mat.in");f>>n;
    for(i=1;i<=n;i++)
        for(j=1;j<=n;j++)
            f>>a[i][j];
    f.close();
}

```

```

}
void afis_sdp(int i,int j)
{
    if (i<=n)
        if (j<=i-1)
            {
                cout<<setw(4)<<a[i][j];
                afis_sdp(i,j+1);
            }
        else
            {
                cout<<endl;
                afis_sdp(i+1,1);
            }
}
void main()
{
    citire(a,n);
    afis_sdp(2,1);
    cout<<endl;
    getch();
}

```

7. De la tastatură se citește un șir a cu n componente numere reale. Să se afișeze, printr-o funcție recursivă, elementele șirului în ordine inversă.

```

#include<conio.h>
#include<iostream.h>
int a[20],n;
void citire(int a[20], int &n)
{
    int i;
    cout<<"n=";cin>>n;
    for(i=1;i<=n;i++)
        {
            cout<<"a["<<i<<"]=";cin>>a[i];
        }
}
void afis_invers(int i)
{
    if (i<=n)
        {
            afis_invers(i+1);cout<<a[i]<<" ";
        }
}

```

```

}
void main()
{
    citire(a,n);
    afis_invers(1);
    cout<<endl;
    getch();
}

```

Probleme propuse

1. Se dă un text. Afișați cuvintele sau cuvântul palindrom folosind o funcție recursiva de verificare a unui cuvânt palindrom.
2. De la tastatură se citește un șir a cu n componente numere reale. Să se afișeze, printr-o funcție fără tip recursivă, elementele șirului în ordinea inversă.
3. Verificați dacă o matrice este triunghiular superioară, față de diagonal principală, folosind o funcție recursivă.
4. Se citește un șir a cu n componente numere reale. Să se inverseze elementele șirului utilizând o funcție fără tip recursivă.
5. Se citesc numere dintr-un fișier de intrare *bac.in*. Verificați printr-o funcție recursiva dacă există cel puțin un număr prim
6. Se consideră șirul de numere naturale definit astfel: 1,1,2,1, 2,3,1,2,3,4,1, Pentru o valoare citită de la tastatură, n, să se afișeze primi n termeni ai șirului.
7. Să se verifice dacă printre n numere citite se găsește un număr care are suma cifrelor un număr prim folosind minim o funcție recursivă.
8. Se dă un șir a cu n componente numere întregi și o valoare k. Să se permute circular elementele șirului cu k poziții la stânga.
9. Fără să se rezolve o ecuație de gradul al doilea cu soluțiile x_1 și x_2 calculati $x_1^n + x_2^n$
10. Se citeste de la tastatură un șir cu n componente numere întregi. Să se returneze prin intermediul a doi parametri suma

elementelor pare, respectiv produsul elementelor impare. Se va utiliza o funcție fără tip recursivă.

11. Să se afișeze toate numerele prime cuprinse între două valori naturale citite a și b.
12. Dându-se o progresie aritmetică prin rație și primul termen să se afișeze primii n termeni utilizând o funcție fără tip recursivă.
13. Câte cifre impare conține un număr natural?
14. Se citește de la tastatură un număr natural n de maxim 9 cifre. Utilizând o funcție fără tip recursivă să se afișeze scrierea numărului ca puteri ale lui 2.
15. Câte componente pare situate pe poziții pare conține un vector de numere naturale
16. Să se afișeze printr-o funcție fără tip recursivă elementele triunghiului lui Pascal până la nivelul p, elemente care reprezintă coeficienții binomului lui Newton din dezvoltarea $(a + b)^n$.
17. Se da un vector de numere întregi. Să se transforme fiecare element din baza 10 în baza 2 folosind o funcție recursivă pentru transformare.
18. Realizați o procedură recursivă care să afișeze triunghiul de numere definit mai jos:

$$\begin{array}{cccccccc}
 1 & 2 & 3 & 4 & \dots & n-2 & n-1 & n \\
 1 & 2 & 3 & 4 & \dots & n-2 & n-1 & \\
 & & & & \dots & & & \\
 & & & & & & 1 & 2 \\
 & & & & & & & 1
 \end{array}$$
19. Se citesc n numere .Determinați perechile de numere prime folosin o funcție pentru calculul cmmdc-ului dintre 2 valori date.
20. Cunoscând termenii șirului Fibonacci, să se descompună numărul natural n în sumă de termeni ai șirului, astfel încât numărul termenilor utilizați să fie minim. Se va utiliza o funcție fără tip recursivă.
21. Calculați suma: $S = -1! + 2! - 3! + \dots + (-1)^n n!$, unde $n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$.

22. Se citește un șir cu n componente. Realizați o funcție fără tip recursivă care să afișeze elementele șirului de pe poziții prime în ordinea inversă din șir
23. Se dă un text de la tastatură. Să se afișeze numărul de vocale conținute în text.
24. Se citesc dimensiunile unei matrici n și m . Realizați o funcție fără tip recursivă care să citească elementele unei matrici.
25. Verificați dacă o cifră k se află printre cifrele unui număr natural dat.
26. Din fișierul text *MAT.IN* se citește o matrice pătratică de dimensiune $n \times n$. Să se afișeze, printr-o funcție fără tip recursivă, elementele de deasupra diagonalei principale, inclusiv diagonală.
27. Calculați C_n^k
28. Din fișierul text *MAT.IN* se citește o matrice pătratică de dimensiune $n \times n$. Să se afișeze, printr-o funcție fără tip recursivă, elementele de sub diagonală secundară, inclusiv diagonală.
29. Afișați toate numerele palindromice de 3 cifre (un număr palindrom este acela pentru care inversul său coincide cu numărul inițial).
30. De la tastatură se citește un șir a cu n componente numere reale. Să se afișeze, printr-o funcție fără tip recursivă, elementele șirului care sunt pătrate perfecte în ordinea inversă apariției în șir.
31. Verificați dacă două cuvinte sunt anagrame
32. De la tastatură se citește un șir a cu n componente numere reale. Să se afișeze, printr-o funcție fără tip recursivă, elementele șirului care sunt pătrate perfecte în ordinea inversă apariției în șir.
33. De la tastatură se citește un număr natural n . Să se determine și să se afișeze numărul obținut prin inversarea cifrelor.
34. Se consideră șirul de numere naturale definit astfel: $1, 1, 2, 3, 2, 1, 1, 2, 3, 4, 5, 4, 3, 2, 1, \dots$. Pentru o valoare, citită de la tastatură, n , să se afișeze primii n termeni ai șirului.
35. Se consideră un șir de numere întregi citite pe rând, până la întâlnirea cifrei 0 care marchează sfârșitul șirului. Să se

afișeze câte numere prime sunt în șir folosind o funcție care să verifice dacă un număr este prim

36. Dându-se o progresie geometrică prin rație și primul termen să se afișeze primii n termeni utilizând o funcție fără tip recursivă.
37. Se citește baza de numerație b ($1 < b < 10$) și un număr n în baza b . Să se determine transformarea numărului din baza b în baza 10.
38. Se dă dimensiunea unei matrici pătratice $n \times n$. Să se umple matricea cu numere naturale având proprietatea că sunt prime. Se va utiliza o funcție fără tip recursivă.
39. Dintre primele n numere naturale, să se afișeze acelea cu proprietatea că suma cifrelor de pe poziții pare lor este impară.
40. Realizați o funcție fără tip recursivă care să afișeze triunghiul de numere definit mai jos:

```

1 1 1 1 ..... 1 1 1
2 2 2 2 ..... 2 2
3 3 3 3 ..... 3
.....
n-2 n-2 n-2
n-1 n-1
n

```

41. Să se afișeze primele q numere care au suma cifrelor mai mică decât o valoare dată m .
42. Se dă un număr natural n format din maxim 9 cifre. Utilizând o funcție fără tip recursivă să se afișeze transformarea numărului în baza 2.
43. Scrieți un program care, pentru doi întregi x și n citiți de la tastatură, afișează valoarea expresiei:
44.
$$s = \frac{x}{1} + \frac{x^3}{3} + \frac{x^5}{5} + \dots + \frac{x^{2n-1}}{2n-1}.$$
45. Se citește un număr natural n . Se citesc n valori succesive, fără a folosi tablouri, să se afișeze numerele formate din maximul de cifre.
46. Să se verifice dacă un număr natural conține o cifră pară

47. Să se afișeze cuvintele de lungime maximă dintre cele n cuvinte citite succesiv de la tastatură, fără a utiliza tablouri, utilizând o funcție fără tip recursivă.
48. Afișati primele n numere prime
49. Din fișierul text MAT.IN se citește o matrice pătratică de dimensiune $n \times n$. Să se afișeze, printr-o funcție fără tip recursivă, elementele de sub diagonala principală, exclusiv diagonala.
50. Afișați toate numerele mai mici sau egale cu n natural pentru care suma cifrelor este pară.
51. Din fișierul text MAT.IN se citește o matrice pătratică de dimensiune n . Să se afișeze, printr-o funcție fără tip recursivă, elementele de deasupra diagonalei secundare, exclusiv diagonala.
52. Afișează primii n termeni ai șirului lui Fibonacci, unde valoarea lui n se transmite ca parametru.
53. Se citește un șir a cu n componente numere reale. Să se inverseze elementele șirului utilizând o funcție fără tip recursivă.

BIBLIOGRAFIE

- **C. Popescu** – *Culegere de probleme de informatică*, Editura Donaris, Sibiu, 2002;
- **G.D. Mateescu, P. Fl. Moraru** – *Informatica pentru liceu și bacalaureat*, Materia de clasa a IX-a, Partea I, Varianta Pascal, Editura Donaris, Sibiu, 2003;
- **D. Popescu Anastasiu** – *Culegere de probleme de informatică pentru gimnaziu și liceu*, Editura All Educational, București, 2000;
- **C. Udrea** – *Pascal, teorie și aplicații*, vol. I, clasa a IX-a, Editura Arves, Craiova, 2001;
- **E. Popescu, E. Ursache, C.C. Neacșu, G. Manafu, D.L. Radu** – *Algoritmi & Limbajul Pascal, teorie și aplicații*, Editura Else, Craiova, 2003;
- **D. Stoilescu, P. Stoilescu** – *Culegere de Pascal pentru gimnaziu*, volumul I, Editura Radial, Galați, 2000;

CUPRINS

PREFAȚĂ

Capitolul 1. ȘIRURI DE CARACTERE

2.1. Probleme rezolvate.....

2.2. Probleme propuse.....

Capitolul 2. ÎNREGISTRĂRI

3.1. Probleme rezolvate.....

3.2. Probleme propuse.....

Capitolul 3. SUBPROGRAME

4.1. Probleme rezolvate.....

4.1. Probleme propuse.....

Capitolul 4. STRUCTURI DE DATE ALOCATE STATIC

4.1. Probleme rezolvate.....

4.1. Probleme propuse.....

Capitolul 5. RECURSIVITATE

6.1.1. Probleme rezolvate.....

6.1.2. Probleme propuse.....

BIBLIOGRAFIE