

1.

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))
    .....
printf("%s",s); | cout<<s;
```

2.

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);
| printf("%s %d",s,strlen(s));
```

3.

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];
| printf("%c%c",s[i],s[x-i-1]);
```

4.

a. ENXAME

b. EAENMX

c. NEEEXMA

d. NEMAXE

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

strlen(strcpy(s,s+2))

(6p.)

5.

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; | printf("%s",a);
```

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; | printf("%s",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 **i** este de tip întreg?

(6p.)

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

8.

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

```
cout<<strchr(s,'a'); | printf("%s",strchr(s,'a')); (4p.)
```

a. 2

b. acalaureat

c. 4

d. bcluret

9.

Ce afișează pe ecran secvența de program scrisă alăturat, în care **i** este o variabilă de tip **char**?

(6p.)

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

10.

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?

(6p.)

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

11.

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;
```

(4p.)

- 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`

12.

Ce afișează pe ecran secvența alăturată de program, unde `a` este o variabilă de tip șir de caractere?

(6p.)

```
strcpy(a,"informatica");
strcpy(a+2,a+5);
cout<<a; | printf("%s",a);
```

13.

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";
cout<<strlen(s); | printf("%d",strlen(s));
for (int i=0;i<strlen(s);i++)
    if (strchr("aeiou",s[i])!=NULL)
        s[i]=s[i]-'a'+'A';
cout<<" "<<s; | printf(" %s",s);
```

14.

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.

(6p.)

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

15.

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?

(4p.)

```
char s[11]="abcduecda";
cout<<strlen(s); | printf("%d",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; | printf(" %s",s);
```

16.

Ce va afișa secvența alăturată știind că variabila `a` memorează un șir cu cel mult 100 de caractere, iar variabilele `i` și `k` sunt de tip întreg?

(6p.)

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

17.

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?

(4p.)

```
i=0; char s[11]="abaemeiut";
cout<<strlen(s); | printf("%d",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; | printf(" %s",s);
```

18.

Care vor fi valorile afișate după executarea secvenței alăturate, dacă variabilele **s1**, **s2** și **s3** sunt de tip șir de caractere? (4p.)

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

a. algoritmritm bioritm ritm b. algoritm bioritm ritmalgorithm

19. c. algoritm bioritm ritmbioritm d. algoritm bioritmritm ritm

Ce va afișa secvența alăturată de program, știind că variabila **a** memorează un șir cu cel mult 100 de caractere, iar variabila **i** este de tip întreg? (6p.)

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

20.

Care din următoarele expresii are valoarea 1 dacă și numai dacă șirul de caractere **s**, de lungime 10, este obținut prin concatenarea a două șiruri identice? (6p.)

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

21.

În secvența 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 aceasta să afișeze șirul de caractere ***nf*rm*t*c***. (6p.)

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

22.

Ce va afișa secvența alăturată de program, știind că variabila **x** memorează un șir cu cel mult 100 de caractere, iar variabila **i** este de tip întreg? (6p.)

```
strcpy(x,"bac2008");
cout<<x<<endl; | printf("%s\n",x);
for(i=0;i<strlen(x);i++)
    if (strchr("0123456789",x[i])==0)
        cout<<x[i]; | printf("%c",x[i]);
```

23.

În secvența 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 aceasta să afișeze doar literele mici și literele mari din șirul de caractere memorat în variabila **a**. (6p.)

```
strcpy(a,"Bac 2007 iulie");
for(i=0;i<strlen(a);i++)
    if(...)
        cout<<a[i];
```

24.

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/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: copii*sunt*la***zoo se afișează

25.

ii
ii
oo

Scrieți un program **C/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.

Exemplu: pentru șirurile **marina** și **elena** se va afișa **na** (10p.)

26.

Scrieți secvența de instrucțiuni **C/C++** care permite afișarea pe ecran a mesajului **Corect** dacă un șir de maximum **100** caractere, memorat de variabila **s**, este palindrom sau mesajul **Inc corect** 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 (4p.)

27.

Care va fi șirul de caractere afișat după executarea secvenței alăturate, în care variabila **s** memorează un șir cu cel mult **5** caractere ? (6p.)

```
char s[]="raton";  
s[1]=s[3];  
cout<<s; | printf("%s",s);
```

28.

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. (10p.)

```
nformatca  
infrmatica  
informtic
```

29.

Scrieți un program **C/C++** care citește de la tastatură două caractere **c1** și **c2**, și un text având cel mult **250** caractere (spații și litere ale alfabetului englez), pe care îl modifică înlocuind toate aparițiile caracterului memorat în **c1** cu cel memorat în **c2** și toate aparițiile caracterului memorat în **c2** cu cel memorat în **c1**. Programul afișează pe linii separate ale ecranului atât textul inițial cât și textul obținut după efectuarea înlocuirilor. (10p.)

Exemplu: dacă pentru **c1** se citește **a**, pentru **c2** se citește **o** iar textul citit este:

hocus pocus preparatus

se va afișa :

hocus pocus preparatus

hacus pacus preporatus

30.

Un șir cu maximum **255** de caractere conține cuvinte separate prin unul sau mai multe spații. Cuvintele sunt formate numai din litere mici ale alfabetului englez. Scrieți un program **C/C++** care citește un astfel de șir și îl afișează modificat, prima și ultima literă a fiecărui cuvânt fiind afișată ca literă mare.

Exemplu: pentru șirul: **maine este proba la informatica** se va afișa:

MainE EstE ProBA LA InformaticA (10p.)

31.

Un șir cu maximum **255** de caractere conține cuvinte separate prin câte un spațiu. Cuvintele sunt formate numai din litere mici ale alfabetului englez. Scrieți un program **C/C++** care citește de la tastatură un astfel de șir și îl afișează pe ecran modificat, inversând prin oglindire doar cuvintele care încep cu vocală, ca în exemplu.

Exemplu: pentru șirul: **maine este proba la informatica** se va afișa:

maine etse proba la acitamrofni (10p.)

32.

Un șir cu maximum **255** de caractere conține cuvinte separate prin caracterul *****. Cuvintele sunt formate numai din litere mici ale alfabetului englez. Scrieți un program **C/C++** care citește un astfel de șir și afișează pe ecran șirul obținut prin eliminarea tuturor aparițiilor primului cuvânt, ca în exemplu.

Exemplu: pentru șirul: **bine*bine*e*foarte*bine*** se va afișa:

****e*foarte**** (10p.)

33.

Scrieți programul **C/C++** care citește de la tastatură un șir de cel mult **40** de caractere, format doar din litere mici ale alfabetului englez, și care afișează pe ecran, pe o singură linie, toate vocalele ce apar în șirul citit. Vocalele vor fi afișate în ordinea apariției lor în șir, separate prin câte un spațiu, ca în exemplu. Șirul citit conține cel puțin o vocală și se consideră ca fiind vocale următoarele litere: **a, e, i, o, u**.

Exemplu: dacă se citește șirul **calculator** atunci pe ecran se va afișa: **a u a o** (10p.)

34.

Scrieți programul C/C++ care citește de la tastatură un șir de cel mult 40 de caractere, format doar din litere ale alfabetului englez, și care afișează pe ecran toate șirurile obținute prin eliminarea succesivă a câte unei singure litere din șirul citit, ca în exemplu. Șirurile obținute se vor afișa câte unul pe câte o linie a ecranului.

Exemplu: dacă se citește șirul **abbc** atunci pe ecran se va afișa:

bbc
abc
abc
abb

(10p.)

35.

Se consideră un text format doar din spații și litere mici ale alfabetului englez, care începe cu o literă și care conține cel puțin o vocală din mulțimea {a, e, i, o, u}. Scrieți programul C/C++ care citește de la tastatură un șir cu cel mult 100 de caractere, ca cel descris mai sus și care determină transformarea acestuia prin înlocuirea fiecărei vocale din text cu litera imediat următoare din alfabet (a se înlocuiește cu b, e se înlocuiește cu f ș.a.m.d.). Programul va afișa pe ecran șirul obținut.

Exemplu: dacă șirul citit este **examen de bacalaureat**, după modificare se afișează:
fxbmfndfbbcblbvrfbt (10p.)

36.

Un cuvânt **s**, de cel mult 20 caractere, format doar din litere mici ale alfabetului englez, conține cel puțin o consoană. Scrieți programul C/C++ care citește de la tastatură cuvântul **s**, construiește în memorie și afișează pe ecran cuvântul obținut prin eliminarea tuturor consoanelor din cuvântul **s**. Se consideră consoană oricare literă care nu se află în mulțimea {a, e, i, o, u}.

Exemplu: dacă se citește cuvântul **bacalaureat**, pe ecran se afișează: **aaauea** (10p.)

37.

Scrieți programul C/C++ care citește de la tastatură un cuvânt **s** de cel mult 20 litere mici ale alfabetului englez, construiește în memorie și afișează pe ecran cuvântul **s** după eliminarea primei și a ultimei vocale. Cuvântul **s** conține cel puțin două vocale. Se consideră vocale literele: **a, e, i, o, u**.

Exemplu: dacă se citește cuvântul **bacalaureat**, pe ecran se afișează: **bcalauret** (10p.)

38.

Se consideră un șir **s** având maximum 52 de caractere, șir ce conține numai litere mici ale alfabetului englez și cifre. Primul caracter al șirului este o literă mică, ultimul caracter al șirului este o cifră și fiecare literă mică din șir este urmată de o cifră nenulă. Scrieți un program C/C++ care citește de la tastatură șirul **s**, apoi construiește și afișează pe ecran un nou șir de caractere, format numai din litere mici ale alfabetului englez, șir construit după următoarea regulă: fiecare literă mică se va repeta de atâtea ori de câte ori o indică cifra situată pe poziția imediat următoare în șirul inițial, ca în exemplu.

Exemplu: dacă se citește de la tastatură șirul **a2b1f2** atunci șirul cerut este **aabff**

39.

Scrieți un program C/C++ care citește de la tastatură un număr natural nenul **n** ($n \leq 100$), apoi **n** șiruri de caractere, fiecare șir având maximum 30 de litere mici ale alfabetului englez și afișează pe ecran câte dintre cele **n** șiruri de caractere sunt formate **numai** din vocale. Se consideră vocale literele **a, e, i, o, u**.

Exemplu: dacă **n=3**, iar șirurile citite sunt date alăturat, atunci programul va afișa pe ecran 1 deoarece șirul **ae** este format

40.

numai din vocale.

(10p.)

arc
ae
creion

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).

(6p.)

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

41.

Se consideră mulțimea vocalelor {a, e, i, o, u}. Scrieți o expresie C/C++ care să fie nenulă dacă și numai dacă variabila **c** de tip **char** este o vocală. (6p.)

42.

Scrieți un program **C/C++** care citește de la tastatură un șir de cel mult **50** de caractere (cifre, litere ale alfabetului englez și spații; șirul conține cel puțin o literă), apoi construiește în memorie și afișează pe ecran șirul de caractere obținut din șirul citit prin eliminarea tuturor caracterelor care nu sunt litere.

Exemplu: dacă se citește șirul: **Voi lua 10 la informatica** atunci se va afișa:

43. **Voilualainformatica** (10p.)

Scrieți un program **C/C++** care citește de la tastatură un șir de cel mult **50** de caractere (litere mici și mari ale alfabetului englez, cifre, puncte, virgule și spații) și afișează pe ecran cifra care apare de cele mai multe ori în șirul citit. Dacă șirul conține mai multe cifre cu număr maxim de apariții, atunci se va afișa cea mai mică dintre acestea. Dacă șirul nu conține cifre, se va afișa pe ecran mesajul **NU**.

Exemplu: dacă se citește șirul:

Voi lua 9,5 la matematica 10 la informatica si 10 la romana

atunci se va afișa cifra **0** (pentru că cifrele **0** și **1** apar de cele mai multe ori în șir și **0** este cea mai mică dintre ele) (10p.)

44.

Scrieți un program **C/C++** care citește de la tastatură un șir de cel mult **50** de caractere (litere mici și mari ale alfabetului englez, cifre și spații) și afișează pe ecran litera mică cel mai des întâlnită în șirul citit. Dacă există mai multe litere mici cu număr maxim de apariții, programul o va afișa pe prima dintre ele în ordine alfabetică. Dacă șirul nu conține litere mici, atunci pe ecran se va afișa mesajul **nu**.

Exemplu: dacă se citește șirul:

mergem la munte

atunci se va afișa: **e** (pentru că literele **e** și **m** apar de cele mai multe ori în șir și **e** este prima dintre ele în ordine alfabetică). (10p.)

45.

Scrieți un program **C/C++** care citește de la tastatură un text de cel mult **255** de caractere, dintre care cel puțin unul este o literă mică a alfabetului englez, și afișează pe ecran, pe o singură linie, despărțite prin câte un spațiu, toate literele mici ale alfabetului englez care apar în text. Fiecare literă va fi afișată o singură dată, în ordinea primei ei apariții în text.

Exemplu: pentru textul:

Calculati valoarea expresiei

Pe ecran se va afișa:

a l c u t i v o r e x p s

46.

(10p.)

Scrieți un program **C/C++** care citește de la tastatură un text format din cel mult **200** de litere ale alfabetului englez, în care cuvintele sunt separate printr-un singur spațiu și afișează pe ecran numărul de cuvinte din textul citit, care au prima respectiv ultima literă vocală. În cazul în care în text nu există un astfel de cuvânt, se va afișa pe ecran mesajul **NU EXISTA**. Se consideră vocală orice literă din mulțimea **{a,A,e,E,i,I,o,O,u,U}**.

Exemplu: dacă textul introdus este:

Eratostene a sugerat ca anii bisecti se repeta la fiecare patru ani

pe ecran se va afișa:

47.

4

(10p.)

Scrieți programul **C/C++** care citește de la tastatură un cuvânt format din cel mult **50** de caractere, litere mari ale alfabetului englez, și afișează pe ecran, fiecare pe câte o linie, toate prefixele acestuia, în ordine crescătoare a lungimilor. Un prefix de lungime **k** al unui cuvânt este un subșir format din primele **k** caractere ale acestuia. (10p.)

Exemplu: dacă se citește cuvântul **BACALAUREAT** se vor afișa prefixele:

B

BA

BAC

BACA

BACAL

BACALA

BACALAU

BACALAU

BACALAU

BACALAU

BACALAU

BACALAU

48.

Scrieți programul C/C++ care citește de la tastatură, separate prin **Enter**, două șiruri de caractere: un șir *s* de maximum 255 de caractere, care pot fi litere ale alfabetului englez și spații, apoi un șir *c* de maximum 20 de caractere. Programul va înlocui în șirul *s* toate aparițiile șirului *c* cu un șir de exact aceeași lungime cu *c*, format doar din caractere *, ca în exemplu. Șirul *s* obținut în urma acestei prelucrări va fi afișat pe ecran. În cazul în care *c* nu apare în *s*, programul va afișa mesajul **NU APARE**.

Exemplu: dacă șirul *s* citit este `Din departare se vede tare` iar *c* este `tare` atunci pe ecran se va afișa: `Din depar**** se vede ****`

(10p.)

49.

Scrieți un program C/C++ care citește de la tastatură un număr natural *n* ($0 < n < 101$) și apoi *n* propoziții. Fiecare propoziție este formată din maximum 255 de caractere care sunt numai litere mici ale alfabetului englez și spații. Știind că fiecare propoziție se termină cu caracterul **Enter**, să se afișeze pe ecran propoziția care are cele mai puține vocale (sunt considerate vocale caracterele *a*, *e*, *i*, *o*, *u*). Dacă două sau mai multe propoziții au același număr de vocale, se va afișa ultima propoziție, în ordinea citirii, care îndeplinește condiția.

Exemplu: pentru *n* = 4 și următoarele propoziții:

```
azi a plouat tare
a fost innorat peste tot
maine va fi soare
acum nu mai vreau sa ploua
se va afișa: maine va fi soare
```

(10p.)

50.

Scrieți programul C/C++ care citește de la tastatură un text de cel mult 50 de caractere, (litere mici ale alfabetului englez și spații), text format din mai multe cuvinte, separate prin câte un spațiu, și afișează pe ecran textul obținut din cel inițial prin transformarea primei litere și a ultimei litere ale fiecărui cuvânt în majusculă, restul caracterelor rămânând nemodificate.

Exemplu: dacă se citește textul `azi este examen de bacalaureat`, se va afișa `AzI EstE ExameN DE BacalaureaT`.

(10p.)

51.

Scrieți programul C/C++ care citește de la tastatură un text cu cel mult 100 de caractere (doar litere mici ale alfabetului englez și spații) și un cuvânt cu cel mult 15 caractere, doar litere mici ale alfabetului englez și care afișează pe ecran șirul obținut prin inserarea în textul inițial a caracterului ? după fiecare apariție a cuvântului citit.

Exemplu: dacă se citește de la tastatură textul `examenul de bacalaureat si examenul de atestat` și cuvântul `examenul` se va afișa:

```
examenul? de bacalaureat si examenul? de atestat
```

(10p.)

52.

Scrieți programul C/C++ care citește de la tastatură un text cu cel mult 100 de caractere (doar litere ale alfabetului englez și spații), construiește în memorie și apoi afișează pe ecran șirul de caractere obținut din șirul inițial, în care se inserează după fiecare vocală caracterul *. Se consideră vocale literele *a*, *e*, *i*, *o*, *u*, *A*, *E*, *I*, *O*, *U*.

Exemplu: dacă se citește de la tastatură textul `Examenul de bacalaureat` se va obține și afișa: `E*xa*me*nu*1 de* ba*ca*la*u*re*a*t`.

(10p.)

53.

Scrieți programul C/C++ care citește de la tastatură un cuvânt cu maximum 20 de litere și minimum o literă, construiește și afișează pe ecran cuvântul obținut din inversarea literelor din prima jumătate a cuvântului cu cele din jumătatea a doua. Literele din fiecare jumătate își păstrează ordinea din cuvântul inițial, ca în exemplu. În cazul în care cuvântul are un număr impar de litere, atunci litera din mijloc va rămâne pe loc.

Exemplu: dacă se citește cuvântul `caiet` se va obține cuvântul `etica`, iar dacă se citește cuvântul `cada` se va obține `daca`.

(10p.)

54.

Un șir de caractere `s` se numește "șablon" pentru un alt șir de caractere `x`, dacă este format din caractere din mulțimea `{*, ?, #}`, are aceeași lungime cu `x` și pe fiecare poziție din `s` în care apare `*` în `x` se găsește o vocală, pe fiecare poziție din `s` în care apare `#` în `x` se găsește o consoană și pe fiecare poziție din `s` în care apare `?` putem avea orice caracter în `x`. Se consideră vocală orice literă din mulțimea `{a, e, i, o, u}`.

Scrieți programul C/C++ care citește de la tastatură două șiruri de caractere, de aceeași lungime, formate din cel mult 200 de litere mici ale alfabetului englez, și afișează pe ecran, un șablon comun celor două șiruri citite, care conține un număr minim de caractere `?`.

55. **Exemplu:** pentru șirurile `diamant` și `pierdut` se afișează `###?#?#` (10p.)

Scrieți un program în limbajul C/C++ care citește de la tastatură un singur șir format din cel mult 20 de caractere care reprezintă numele și prenumele unei persoane. Între nume și prenume se află un număr oarecare de caractere spațiu (cel puțin unul). Atât numele cât și prenumele sunt formate numai din litere ale alfabetului englez. Programul construiește în memorie și afișează pe ecran un al doilea șir de caractere, care să conțină prenumele, urmat de exact un spațiu și apoi numele din șirul citit inițial.

Exemplu: dacă se citește șirul:

```
Popescu Vasile
```

se va construi și apoi se va afișa pe ecran șirul

56. Vasile Popescu (10p.)

Scrieți un program în limbajul C/C++ care citește de la tastatură un singur șir, format din cel mult 20 de caractere, care reprezintă numele și prenumele unei persoane. Între nume și prenume se află un număr oarecare de caractere spațiu (cel puțin unul). Atât numele, cât și prenumele, sunt formate numai din litere mici ale alfabetului englez. Programul construiește în memorie și afișează pe ecran un alt șir de caractere, care să conțină inițiala prenumelui (prima literă a prenumelui), urmată de un caracter punct, de exact un spațiu și de numele din șirul citit inițial. Toate literele din șirul afișat vor fi, de asemenea, litere mici.

Exemplu: dacă se citește șirul:

```
popescu vasile
```

se va construi și apoi se va afișa pe ecran șirul

```
v. popescu
```

(10p.)

57.

Scrieți un program în limbajul C/C++ care citește de la tastatură două șiruri, formate fiecare din cel mult 20 de caractere. Primul șir reprezintă numele unei persoane, iar al doilea șir reprezintă prenumele aceleiași persoane. Atât numele cât și prenumele sunt formate numai din litere ale alfabetului englez și fiecare conține cel puțin o consoană. Programul construiește în memorie și afișează pe ecran un al treilea șir de caractere, care conține consoanele din prenumele citit dispuse în ordinea în care apar în prenume urmate de exact un spațiu și de numele citit.

Exemplu: dacă primul șir citit este `Popescu`, iar al doilea este `Vasile`

se va construi și apoi se va afișa pe ecran șirul

58. `Vs1 Popescu` (10p.)

Scrieți programul C/C++ care citește de la tastatură un text de cel mult 50 de caractere, (litere mici ale alfabetului englez și spații), text format din mai multe cuvinte, separate prin câte un spațiu, și afișează pe ecran textul obținut din cel inițial prin transformarea primei litere și a ultimei litere ale fiecărui cuvânt în majusculă, restul caracterelor rămânând nemodificate.

Exemplu: dacă se citește textul `azi este examen de bacalaureat`, se va afișa `Azi`

`EstE ExameN DE BacalaureaT`.

(10p.)

59.

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

60. `strcpy(s, strstr(s, "TI"));` (6p.)

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);`

(6p.)