



Lucrare semestrială la **INFORMATICĂ intensiv**  
An școlar 2018-2019, semestrul I, clasa a X-a  
17-decembrie-2018

Numărul I

1 Subprogramul **cifreImpare** are un singur parametru, **n**, prin care primește un număr natural cu toate cifrele nenule ( $1 < n < 10^9$ ). Subprogramul returnează numărul obținut prin eliminarea tuturor cifrelor impare din **n**, respectiv **-1** dacă nu există astfel de cifre sau dacă toate cifrele lui **n** sunt impare.

Scrieți definiția completă a subprogramului.

Exemplu: dacă **n=23541** subprogramul returnează **24**, iar dacă **n=28** subprogramul returnează **-1**.

Se consideră subprogramul **divizori**, cu trei parametri:

- 2
- **n**, prin care primește o valoare naturală ( $0 < n < 10^6$ );
  - **mic** și **mare**, prin care furnizează cel mai mic și respectiv cel mai mare divizor propriu pozitiv al lui **n** (diferit de 1 și de **n**) sau valoarea 0, prin amândoi, dacă numărul nu are cel puțin doi astfel de divizori.

Scrieți definiția completă a subprogramului.

Exemplu: pentru numărul **n=12**, în urma apelului, **mic=2** și **mare=6**, iar pentru numărul **n=9**, în urma apelului, **mic=0** și **mare=0**.

3 În secvența de instrucțiuni de mai jos variabilele **s1** și **s2** memorează câte un șir cu cel mult 20 de caractere. Scrieți ce se afișează pe ecran în urma executării secvenței.

```
strcpy(s1, "bacalaureat"); cout << strlen(s1);  
strcpy(s2, s1+5); s2[3] = '\0'; cout << s2;
```

4 Fiind date două șiruri de caractere **a** și **b**, îl numim pe **a** **prefix** al lui **b** dacă **a** este egal cu **b** sau dacă **b** se poate obține din **a** prin alipirea la dreapta a unor noi caractere. Variabilele **a** și **b** pot memora câte un șir cu cel mult 20 de caractere. Știind că variabila **b** a fost inițializată cu un șir format dintr-un număr par de caractere, scrieți o secvență de instrucțiuni în urma executării căreia variabila **a** să memoreze un prefix al lui **b** a cărui lungime să fie jumătate din lungimea lui **b**.

Exemplu: dacă **b** memorează șirul **aurari**, atunci **a** memorează șirul **aur**.

5 Un text are cel mult 100 de caractere, iar cuvintele sale sunt formate doar din litere mici ale alfabetului englez și sunt separate prin câte un spațiu.

Scrieți un program **C/C++** care citește de la tastatură un text de tipul precizat mai sus și îl transformă în memorie prin înlocuirea fiecărui cuvânt format din număr par de litere cu simbolul **#**. Programul afișează pe ecran textul obținut sau mesajul **nu exista** dacă textul citit nu conține astfel de cuvinte.

Exemplu: pentru textul **anii de liceu sunt foarte frumoși**  
se afișează **# # liceu # # frumoși**

6 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' ;  
  }  
}
```

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

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

7 Descrieți, exemplificați pe tabloul (7, 8, 2, 5, 3, 9, 4, 6, 1) și apoi implementați algoritmul de sortare prin interclasare (**MergeSort**).



Lucrare semestrială la INFORMATICĂ intensiv  
An școlar 2018-2019, semestrul I, clasa a X-a  
17-decembrie-2018

Numărul II

1	<p>Subprogramul <b>Cifre</b> are un singur parametru, <b>n</b>, prin care primește un număr natural (<math>n \in [0, 10^9]</math>). Subprogramul returnează numărul cifrelor care apar o singură dată în scrierea lui <b>n</b>. Scrieți definiția completă a subprogramului. Exemplu: dacă <b>n=9272017</b>, subprogramul returnează valoarea <b>3</b>.</p>	
2	<p>Un interval cu proprietatea că există un singur număr natural, <b>n</b> (<math>2 \leq n</math>), pentru care valoarea produsului <math>1 \cdot 2 \cdot 3 \cdot \dots \cdot n</math> aparține acestui interval este numit <b>interval factorial</b> al lui <b>n</b>. Exemplu: <b>[5,8]</b> și <b>[3,23]</b> sunt intervale factoriale ale lui <b>3</b>, dar <b>[1,15]</b> și <b>[7,10]</b> nu sunt intervale factoriale ale niciunui număr.</p> <p>Se consideră subprogramul <b>interval</b>, cu trei parametri:</p> <ul style="list-style-type: none"><li>• <b>n</b>, prin care primește un număr natural din intervalul <b>[2,10]</b>.</li><li>• <b>a</b> și <b>b</b>, prin care furnizează câte un număr natural, astfel încât expresia <b>b-a</b> să aibă valoare maximă, iar <b>[a,b]</b> să fie interval factorial al lui <b>n</b>. Scrieți definiția completă a subprogramului.</li></ul> <p>Exemplu: dacă <b>n=3</b>, după apel <b>a=3</b> și <b>b=23</b>.</p>	
3	<p>Variabilele <b>s</b> și <b>x</b> permit memorarea câte unui șir cu maximum <b>20</b> de caractere. Fără a utiliza alte variabile, scrieți o secvență de instrucțiuni <b>C/C++</b> în urma executării căreia variabila <b>x</b> să permită memorarea șirului corespunzător variabilei <b>s</b>, dacă el are un număr par de caractere, sau a unui șir obținut din acesta prin eliminarea caracterului din mijloc, în caz contrar, ca în exemplu. Exemplu: dacă șirul corespunzător variabilei <b>s</b> este <b>pictura</b> șirul memorat prin intermediul variabilei <b>x</b> este <b>picura</b></p>	
4	<p>În secvența de instrucțiuni de mai jos variabilele <b>s1</b> și <b>s2</b> memorează câte un șir cu cel mult <b>20</b> de caractere. Scrieți ce se afișează pe ecran în urma executării secvenței. <b>strcpy(s1, "informatica"); cout&lt;&lt;strlen(s1);</b> <b>strcpy(s2, "mate"); strcat(s2, strstr(s1, "ma"));</b> <b>cout&lt;&lt;s2;</b></p>	
5	<p>Se consideră un text cu cel mult <b>100</b> de caractere (litere mici ale alfabetului englez și spații), în care cuvintele sunt separate prin câte un spațiu. Înaintea primului cuvânt și după ultimul cuvânt nu există spații. Scrieți un program <b>C/C++</b> care citește de la tastatură un text de tipul menționat mai sus și determină transformarea acestuia în memorie prin înlocuirea fiecărui cuvânt format din trei litere cu simbolul <b>*</b>. Programul afișează pe ecran textul obținut. Exemplu: pentru textul <b>bun este izvorul ce are apa rece</b> se afișează <b>* este izvorul ce * * rece</b></p>	
6	<p>Pentru definiția alăturată a subprogramului <b>f</b>, ce se afișează ca urmare a apelului <b>f(15,2);</b> <b>void f (int n, int x)</b> <b>{ if(x&gt;n)cout&lt;&lt;0;</b> <b>  else</b> <b>    if(x%4&lt;=1) f(n,x+1);</b> <b>    else</b> <b>      { f(n,x+3);</b> <b>        cout&lt;&lt;1;</b> <b>      }</b> <b>}</b></p>	<p>Pentru subprogramul <b>suma</b> definit alăturat, scrieți valoarea expresiei <b>suma(5,4)</b>.  <b>int suma (int a,int b)</b> <b>{</b> <b>  if (a==0 &amp;&amp; b==0) return 0;</b> <b>  else if (a==0) return 1+suma(a,b-1);</b> <b>  else return 1+suma(a-1,b);</b> <b>}</b></p>
7	<p>Descrieți, exemplificați pe tabloul <b>(7, 2, 5, 3, 4)</b> și apoi implementați algoritmul de sortare rapidă (<b>QuickSort</b>).</p>	



Lucrare practică la INFORMATICĂ  
An școlar 2018-2019, semestrul I  
20-decembrie-2018

Numărul I

Se dă un tablou cu  $n$  elemente, numere naturale. Să se înlocuiască fiecare element din tablou care nu este număr prim cu cel mai mic număr prim, mai mare decât el.

Se vor defini și apela următoarele subprograme:

- *citire*, care citește de la tastatură valoarea lui  $n$  și cele  $n$  elemente ale tabloului
- *afisare*, care afișează pe ecran elementele tabloului, separate prin exact un spațiu
- *prim*, care verifică dacă un număr natural este prim
- *urmatorul\_prim*, care determină pentru un număr dat cel mai mic număr prim, mai mare decât acesta, folosind subprogramul *prim*
- *inloc*, care realizează înlocuirile cerute.

Date de intrare

Se citește de la tastatură numărul  $n$ , iar apoi cele  $n$  elemente ale tabloului.

1 Date de ieșire

(3p) Se vor afișa pe ecran elementele tabloului, separate prin exact un spațiu.

Restricții și precizări

- $1 \leq n \leq 1000$
- elementele tabloului vor fi mai mici decât  $1.000.000.000$

Exemplu

**Intrare**

```
7
7 18 18 5 14 20 4
```

**Ieșire**

```
7 19 19 5 17 23 5
```

Se consideră un text format din cel mult 200 de caractere litere mici și spații. Cuvintele din text sunt separate prin unul sau mai multe spații. Scrieti un program care citește textul și care:

- determină numărul de consoane din textul citit;
- înlocuiește prima literă a fiecărui cuvânt din textul citit cu litera X (mare);
- modifică textul citit prin aranjarea în ordine lexicografică a tuturor cuvintelor din text și separarea lor prin câte un singur spațiu.

Date de intrare

Fișierul de intrare *prosirx.in* conține pe prima linie textul.

Date de ieșire

Fișierul de ieșire *prosirx.out* va conține:

- pe prima linie răspunsul la cerința a)
- pe a doua linie răspunsul la cerința b)
- pe a treia linie răspunsul la cerința c)

2 (4p)

Restricții și precizări

- textul este format din cel mult 200 de caractere, doar litere mici sau spații

Exemplu

*prosirx.in*

```
oana este la scoala
```

*prosirx.out*

```
7
Xana Xste Xa Xcoala
este la oana scoala
```



Lucrare practică la INFORMATICĂ  
An școlar 2018-2019, semestrul I  
20-decembrie-2018

Numărul II

Se dă un tablou cu  $n$  elemente, numere naturale. Să se elimine din tablou toate elementele care sunt palindrom.

Se vor defini și apela următoarele subprograme:

- *citire*, care citește de la tastatură valoarea lui  $n$  și cele  $n$  elemente ale tabloului
- *afisare*, care afișează pe ecran elementele tabloului, separate prin exact un spațiu
- *palindrom*, care verifică dacă un număr dat ca parametru este palindrom
- *eliminare*, care elimină din tablou un element a cărui poziție este dată ca parametru.

Date de intrare

Se citește de la tastatură numărul  $n$ , iar apoi cele  $n$  elemente ale tabloului.

Date de ieșire

Se vor afișa pe ecran elementele tabloului, separate prin exact un spațiu.

1  
(4p)

Restricții și precizări

- $1 \leq n \leq 1000$
- elementele tabloului vor fi mai mici decât  $1.000.000.000$

Exemplu

**Intrare**

```
7
17 181 22 56 4 20 420
```

**Ieșire**

```
17 56 20 420
```

Se consideră un text format din cel mult  $200$  de caractere litere mici și spații. Cuvintele din text sunt separate prin unul sau mai multe spații. Scrieti un program care citește textul și care:

- determină numărul de vocale din textul citit;
- înlocuiește ultima literă a fiecărui cuvânt din textul citit cu litera Z (mare);
- modifică textul citit prin aranjarea în ordine invers lexicografică a tuturor cuvintelor din text și separarea lor prin câte un singur spațiu.

Date de intrare

Fișierul de intrare *prosirz.in* conține pe prima linie textul.

Date de ieșire

Fișierul de ieșire *prosirz.out* va conține:

- pe prima linie răspunsul la cerința *a*)
- pe a doua linie răspunsul la cerința *b*)
- pe a treia linie răspunsul la cerința *c*)

4  
(4p)

Restricții și precizări

- textul este format din cel mult  $200$  de caractere, doar litere mici sau spații

Exemplu

*prosirz.in*

```
oana este la scoala
```

*prosirz.out*

```
9
oanZ estZ lZ scoalZ
scoala oana la este
```