

Examen de admitere sesiunea iulie 2017
Informatică

Toate subiectele sunt obligatorii.
Se acordă 10 puncte din oficiu.
Timpul efectiv de lucru este de 3 ore.

SUBIECTUL I (30 puncte)

Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. (5p.) Pentru care din valorile de mai jos expresia $((y-3) * (y+3) > 0) \ \&\& \ (\sim(x >> 3))$ are valoarea **TRUE** (1), considerând că variabilele **x** și **y** sunt de tip **int** ?
- a) $x = 12, y = 3$
 - b) $x = 9, y = -3$
 - c) $x = 9, y = 4$
 - d) $x = 12, y = -4$

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

2. (25p.) Se consideră algoritmul următor, reprezentat în pseudocod. S-a notat cu **a%b** restul împărțirii numărului natural **a** la numărul natural nenul **b**.

```
citește a,b
    (a,b numere naturale nenule)
r <-- a%b
cât timp r<>0 execută
    | a = b
    | b = r
    | r = a%b
-#
scrie b
```

- a) (5p.) Scrieți valorile afișate la ieșire, dacă de la intrare se citesc numerele **15, 21**.
- b) (5p.) Scrieți cele mai mici valori de intrare pentru ca la ieșire să se afișeze valoarea **9**.
- c) (5p.) Scrieți în pseudocod un algoritm echivalent cu cel dat (adică determină același rezultat ca și algoritmul inițial), înlocuind structura repetitivă de tip **cât timp...execută** cu o altă structură repetitivă.
- d) (10p.) Scrieți programul în limbajul C / C++ / Pascal corespunzător algoritmului dat.

SUBIECTUL al II-lea (30 puncte)

Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. (5p.) Se consideră funcția:

Limbaajul C:

```
int f(int n){
    printf("%d", n%10);
    return n/10;
}
```

Limbaajul Pascal:

```
function f(n:integer):integer;
begin
    write(n mod 10);
    f := n div 10;
end;
```

Ce va afișa apelul:

```
printf("%d", f(f(f(1423))));
```

```
write(f(f(f(1423))));
```

- a) 142
- b) 3241
- c) 1423
- d) 324

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

2. (10p.) De pe mediul de intrare se citește un număr întreg pozitiv (reprezentabil prin tipul **long int**) cuprins în intervalul $[0, 2000000000]$. Să se realizeze programul în C/C++/Pascal care verifică dacă numărul citit este palindrom. În cazul în care numărul este palindrom se va afișa mesajul "Este palindrom" altfel se va afișa mesajul "Nu este palindrom". Un număr este palindrom dacă se citește la fel de la stânga la dreapta și de la dreapta la stânga. De exemplu, un astfel de număr este 4315134. Pentru rezolvarea problemei nu se vor folosi tablouri de variabile pentru păstrarea cifrelor numărului.

3. (15p.) Se consideră o matrice pătrată A cu n linii și n coloane, cu elemente numere întregi cuprinse în intervalul $[0, 100]$. Se cere parcurgerea în spirală a matricei A , adică:

	1	2	...	n				
1	●	→	→	→	→	→	→	●
2	●	→	→	→	→	→	●	↓
	↑	●	→	→	→	●	↓	↓
	↑	↑	↓	↓	↓
⋮	↑	↑	↑	↓	↓
	↑	↑	●	...	←	●	↓	↓
	↑	●	←	←	←	←	●	↓
n	●	←	←	←	←	←	←	●

De exemplu, pentru matricea A cu 4 linii și 4 coloane:

```
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
```

parcurgerea în spirală presupune afișarea elementelor matricei în ordinea următoare:

```
1, 2, 3, 4, 8, 12, 16, 15, 14, 13, 9, 5, 6, 7, 11, 10
```

a) (11p.) Scrieți algoritmul de rezolvare a problemei în pseudocod.

b) (4p.) Descrieți rolul tuturor variabilelor din algoritm și indicați datele de intrare și datele de ieșire.

SUBIECTUL al III-lea (30 puncte)

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

1. (5p.) În secvența de instrucțiuni de mai jos variabilele n , m , i , j și x sunt de tip întreg, iar v este un tablou unidimensional de numere întregi cu indici de la 0 la 10:

a) *Limbaajul C/C++*

```
i = 0;
n = 19;
while (n>0) {
    i++;
    v[i] = n%2;
    n = n/2;
}
x = v[1];
for (j=1;j<i;j++) {
    v[j] = v[j+1];
}
v[i] = x;
m = v[i];
for (j=2;j<=i;j++) {
    m = .....;
    if (v[i+1-j]+v[i+2-j]>0)
        printf("%d\n",m);
}
```

b) *Limbaajul Pascal*

```
i := 0;
n := 19;
while n>0 do begin
    i := i+1;
    v[i] := n mod 2;
    n := n div 2
end;
x := v[1];
for j := 1 to i-1 do begin
    v[j] := v[j+1];
end;
v[i] := x;
m := v[i];
for j := 2 to i do begin
    m = .....;
    if v[i+1-j]+v[i+2-j]>0 then
        writeln(m)
end;
```

Indicați o expresie care poate înlocui punctele de suspensie astfel încât, în urma executării secvenței, să se afișeze pe ecran valorile următoare:

3
6
25

2. (5p.) Să se determine și să se afișeze cel mai mic termen al șirului lui Fibonacci ($f_1 = 1$, $f_2 = 1$, $f_i = f_{i-1} + f_{i-2}$, $i > 2$), strict mai mare decât 2017, precum și indicele acestuia, în această ordine.

3. (10p.) O mulțime de $2 \leq n \leq 10$ prieteni doresc să stabilească o întâlnire într-o anumită zi a săptămânii. Fiecare dintre ei este disponibil numai într-un anumit interval orar $[x_i, y_i]$ unde valorile x_i și y_i pentru i de la 1 la n sunt numere întregi ce satisfac condițiile $7 \leq x_i < y_i \leq 22$. Acest lucru înseamnă că prietenul i este disponibil să se întâlnească la orice moment de timp din intervalul închis $[x_i, y_i]$. Se cere să se scrie un program C/C++/Pascal care să determine dacă cei n prieteni se pot întâlni toți în ziua respectivă. Programul va citi valoarea lui n , urmată de valorile x_i și y_i pentru toți i de la 1 la n și va afișa șirul de caractere „DA” dacă întâlnirea este posibilă, respectiv șirul de caractere „NU” dacă întâlnirea nu este posibilă.

4. (10p.) Se consideră un fișier text de intrare, **input.txt**, ce conține două linii: pe prima linie se specifică un număr întreg pozitiv $n < 10000$, iar pe a doua linie se specifică un șir de n numere întregi, a_1, a_2, \dots, a_n , separate de câte un spațiu. Numerele sunt în intervalul $[1, 100]$. Se cere să se determine cele mai lungi două subșiruri de numere succesive din șirul dat, astfel încât elementele primului subșir să fie toate egale, respectiv elementele celui de al doilea subșir să fie distincte oricare două între ele. Să se scrie aceste subșiruri într-un fișier text de ieșire, **output.txt**, pe linii separate. Elementele fiecărui subșir vor fi separate prin câte un spațiu. În cazul în care există mai multe subșiruri care îndeplinesc fiecare dintre condițiile problemei atunci se vor determina și afișa primele dintre aceste subșiruri care apar în fișierul de ieșire. Spre exemplu, dacă șirul de intrare are 20 de elemente 2 4 4 4 5 6 7 23 7 34 34 34 23 23 10 10 1 1 1 23, atunci se vor afișa pe câte o linie separată 4 4 4 și 4 5 6 7 23.