



## Examen PC – 12 februarie 2023

1p din oficiu

1. (2p) Scrieți în ordine, pe 8 rânduri separate, rezultatele afișate la ieșirea standard obținute în urma execuției programului C de mai jos:

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

#define PATRAT(x) (x*(x))
float x;

float f() {
    static float y=3.2;
    x += ++y;
    return x;
}

void g(float* t) {
    int v;
    while (v=t[0]) {
        if (v>100) {
            t++;
            continue;
        }
        t[0] *= 10;
    }
}

int main()
{
    int a;
    if (a>=0)
        a -= a+1;
    printf("%f\n", a/x);
}
```

```
char c=128;
int d=c+5;
printf("%d\n", d);

int x=7;
int y=2;
printf("%d\n", PATRAT(x-y));

char *v="";
printf("%s\n", (v=='\0')?"ZERO":"NUL");

f();
float z=f();
printf("%f\n", z);

float t[] = {1.06f, 102.3f, 0.9f, 2};
g(t);
printf("%f\n", t[0]+t[1]+t[2]+t[3]);

struct data {
    char r[4];
    int t;
}s;
printf("%d\n", sizeof(s));

memset(&s, 10, sizeof(s));
printf("%X", *s.r + s.t);
return 0;
}
```

2. (2p) Scrieți în C o funcție *data* care citește de la intrarea standard o dată calendaristică de forma *zz.ll.aaaa* unde: *zz* este ziua și trebuie să fie o valoare între 1 și 31, *ll* este luna și trebuie să fie o valoare între 1 și 12, iar *aaaa* este anul și trebuie să fie o valoare între 1900 și 2100. Scrieți apoi un program care utilizează funcția *data* pentru a citi două date calendaristice și care afișează apoi acea dată care este calendaristic prima. Dacă datele de intrare nu respectă specificațiile de mai sus, se va afișa la ieșirea standard un mesaj de eroare. Observație: programul nu trebuie să valideze dacă o anumită zi există într-o anumită lună dintr-un anumit an (exemplu 29.02.2022)!

Exemple:

Citire de la intrarea standard:	Afișare la ieșirea standard:
14.08.2022 23.07.2022	23.07.2022 este calendaristic prima
14.08.2022 23.07.2023	14.08.2022 este calendaristic prima
14.13.2000 32.07.2022	Date invalide!
14.08. 23/07/2022	Date invalide!

3. (2p) Scrieți în C o funcție recursivă numită *turn* care primește ca și parametri un șir de caractere *s* memorat într-un tablou (și care conține cel puțin un caracter) și o înălțime *h* (număr natural pozitiv) și care verifică dacă toate caracterele șirului *s* formează, în ordine, un turn de înălțime *h*, care are pe fiecare nivel *x* același caracter (care apare de exact *x* ori). În caz afirmativ funcția trebuie să returneze valoarea 1, iar în caz negativ 0. Observații: Turnul format trebuie să aibă exact *h* niveluri și să conțină toate caracterele din *s*. Formarea turnului începe de la nivelul de bază (jos) unde se pun ultimele *h* caractere din *s*, apoi se continuă cu următoarele *h-1* caractere din *s* pe următorul nivel, în sus,... și în același mod până se ajunge la vârful turnului.

Exemple:

Înainte de apel:	Turnul:	Apelul funcției:	După apelul funcției:
<code>s ← "EXXAAAMMM"</code> <code>h ← 4</code>	E XX AAA MMMM	<code>x ← turn(s, h)</code>	<code>x → 1</code>
<code>s ← "EXXABAMMM"</code> <code>h ← 4</code>	E XX ABA MMMM	<code>x ← turn(s, h)</code>	<code>x → 0</code>
<code>s ← "EAAAMMM"</code> <code>h ← 4</code>	E AAA MMMM	<code>x ← turn(s, h)</code>	<code>x → 0</code>
<code>s ← "A"</code> <code>h ← 1</code>	A	<code>x ← turn(s, h)</code>	<code>x → 1</code>
<code>s ← "WR"</code> <code>h ← 1</code>	R	<code>x ← turn(s, h)</code>	<code>x → 0</code>

4. (3p) În fișierul text cu numele trimis ca și prim argument la execuția programului sunt scrise pe prima linie, separate printr-un spațiu, numărul de rânduri (maxim 10000) și numărul de coloane (maxim 10000) ale unei matrice cu caractere care definește o *hartă*; pe următoarele linii, în fișier, sunt scrise (fără spații) elementele matricei (litere mari). Urmează scrise în fișier numărul de rânduri (maxim 30) și numărul de coloane (maxim 30) ale unei matrice cu caractere care definește un *șablon*; pe următoarele linii, în fișier, sunt scrise elementele matricei (litere mari și caractere punct '.'). Caracterele punct '.' reprezintă spațiu liber în șablon. Scrieți un program C, structurat în mai multe funcții, care citește toate datele din fișierul text de intrare și scrie într-un fișier binar de ieșire numărul de locații unde se potrivește întreg *șablonul* pe suprafața *hărții*. Șablonul se potrivește doar dacă toate elementele acestuia coincid cu cele de pe hartă. Se presupune că datele existente în fișierul de intrare sunt valide. Afișați un mesaj de eroare în cazurile când nu se poate citi/scrie din/în fișiere.

Exemplu:

Conținutul fișierului text de intrare (locurile unde se potrivește <i>șablonul</i> pe suprafața <i>hărții</i> sunt evidențiate prin litere îngroșate):	Rezultat scris în fișierul binar de ieșire:
5 9 AERDIJMKL POW <b>EXJE</b> FK DLOK <b>AMGN</b> H L <b>EXTE</b> DLLP DQ <b>AMHN</b> NTD 2 5 <b>EX.E.</b> <b>.AM.N</b>	2