

Examen PC – 8 februarie 2020

1p din oficiu

1. (2,5p) Scrieți în ordine, pe 10 rânduri separate, rezultatele afișate la ieșirea standard obținute în urma execuției programului C de mai jos știind că acesta este executat dintr-o consolă cu următoarele argumente:
 ...>program.exe Sesiune de iarna

<pre>#include <stdio.h> #include <string.h> #define INVERS(x) (1.0/x) int g; struct student { float varsta; char nume[12]; }p,r; void seteaza(struct student* s, char* t) { static float v=22; strcpy(s->nume, t); (*s).varsta=++v; g-=5; } int main(int argc, char* argv[]) { unsigned char a=100; unsigned char b=28; printf("%f\n",INVERS(a-b));</pre>	<pre> unsigned char c=a+b; printf("%d\n",c); printf("%d\n",a&b); int x=0; switch(g) { case '0': x++; case '\0': x+=20; default: x+=500; } printf("%d\n",x); printf("%d\n",sizeof(p)); seteaza(&p,"Alexandru"); r=p; seteaza(&r,"Emma"); printf("%s\n",p.nume); printf("%f\n",(&r)->varsta); printf("%x\n",g); enum {zece, douazeci=20, treizeci}; printf("%d\n",zece+treizeci); printf("%s\n",argv[2]); return 0; }</pre>
---	---

2. (1,5p) Scrieți un program C care citește mai întâi de la intrarea standard două momente de timp în format HH:MM:SS unde HH=ora (00-24), MM=minute (00-59), SS=secunde (00-59). Verificați ambele momente de timp dacă sunt introduse în formatul specificat iar în caz negativ afișați un mesaj de eroare. În cazul în care momentele de timp sunt valide determinați și afișați la ieșirea standard timpul scurs de la primul moment de timp până la întâlnirea celui de-al doilea, exprimat în ore, minute și secunde. Observație: cel de-al doilea moment de timp poate să aparțină zilei următoare.

Exemple:

Citire de la intrarea standard:	Afișare la ieșirea standard:
23:26:01 01:04:15	1 ore, 38 minute si 14 secunde
07:34:29 10:02:11	2 ore, 27 minute si 42 secunde
21:01:54 02:37:59	5 ore, 36 minute si 5 secunde
04:01:54 12:61:79	Date de intrare invalide!

3. (1,5p) Scrieți în C funcția *palindrom* care verifică dacă un șir de caractere primit ca și parametru este sau nu palindrom. Scrieți apoi în C funcția *sir_palindromuri* care formează un șir cu toate cuvintele palindrom existente într-o matrice de cuvinte dată. Funcția *sir_palindromuri* primește ca și argumente de intrare: matricea de cuvinte, numărul de linii și coloane ale acesteia. Funcția *sir_palindromuri* returnează șirul de palindromuri format, iar prin intermediul ultimului argument de ieșire numărul de elemente ale acestuia. Nu este permisă utilizarea de variabile globale.

Exemple:

Înainte de apel:	Apelul funcției:	După apel:
<pre>mat←["ac" "lup"] ["ou" "lupul"] ["ABBA" "greu"] m←3 n←2</pre>	<code>x←sir_palindromuri(mat,m,n,nr)</code>	<pre>x→["lupul" "ABBA"] nr→2</pre>
<pre>mat←["verde" "rosu" "alb"] ["gri" "alb" "roz"] m←2 n←3</pre>	<code>x←sir_palindromuri(mat,m,n,nr)</code>	<pre>x→NULL nr→0</pre>

4. (1,5p) Scrieți în C o funcție recursivă numită *putere* care verifică dacă un număr natural nenul n este putere (număr natural) a unui alt număr natural nenul a . Cele două valori n și a sunt trimise ca și argumente la apelul funcției. Funcția trebuie să returneze valoarea 1 în caz afirmativ și 0 în caz negativ.

Exemple:

Înainte de apel:	Apelul funcției:	După apel:
<pre>n←81 a←3</pre>	<code>x←putere(n,a)</code>	<code>x→1</code>
<pre>n←108 a←3</pre>	<code>x←putere(n,a)</code>	<code>x→0</code>
<pre>n←125 a←5</pre>	<code>x←putere(n,a)</code>	<code>x→1</code>
<pre>n←2 a←5</pre>	<code>x←putere(n,a)</code>	<code>x→0</code>
<pre>n←1 a←20</pre>	<code>x←putere(n,a)</code>	<code>x→1</code>
<pre>n←20 a←1</pre>	<code>x←putere(n,a)</code>	<code>x→0</code>

5. (2p) Fișierul text *siruri.txt* conține un număr de maximum 1000 de șiruri cu numere reale. Fiecare șir poate conține teoretic un număr infinit de elemente. Elementele fiecărui șir sunt scrise pe linii separate în fișier, iar valorile componente sunt separate printr-un singur spațiu. Scrieți un program C, structurat în funcții, care citește datele din fișierul de intrare și care determină și scrie în fișierul text de ieșire cu numele *ordonate.txt*, pentru fiecare șir în parte, dacă elementele acestuia sunt sau nu ordonate (fie în ordine crescătoare fie în ordine descrescătoare). Se presupune că datele scrise 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 <i>siruri.txt</i> :	Rezultate scrise în fișierul de ieșire <i>ordonate.txt</i> :
<pre>-8.456 2.74 20 20.1 20.1 150.4598 -8.456 2.74 20 10.34 45.34 56.78987 -345.6543 984.45545 652.3544 652.3544 100.2 -74 1000.23 1000 999.898 1000.3 998.1</pre>	<pre>DA NU DA DA NU</pre>