

## Examen la Programarea Calculatoarelor - subiect A

1. feb. 2024 - sesiunea iarnă

1. (1p) Ce este depășirea limitei de reprezentare a datelor? Dați două exemple diferite și explicați!
2. (2p) Scrieți în ordine, pe 8 rânduri separate, rezultatele afișate la ieșirea standard obținute în urma execuției programului C de mai jos:

<pre>#include &lt;stdio.h&gt; #include &lt;string.h&gt; #include &lt;stdlib.h&gt;  char E; #ifdef E     char c = 127; #else     char c = 125; #endif  void f(float *t) {     *t = 200;     t = (float*)calloc(3, sizeof(float)); }  int main() {     int x=15;     switch(c) {         case 0 : x+=10;         case 127: x*=3;         case 125: x/=2;         default : x++;     }     printf("%d\n",x);      while (c%10)         c++;     printf("%d\n",c);      int a=10, b=20;     int r = (a=40)    (b=70);     printf("%d\n",a+b+r);</pre>	<pre>int contor=0; for (int i=0; i&lt;10; i++)     for (int j=0; j&lt;20; j++)     {         if (j&gt;5)         {             break;             break;         }         contor++;     } printf("%X\n",contor);  float t[]={1.01, 2.02, 3.03}; f(t); printf("%f\n",t[0]+t[1]+t[2]);  char s[] = "MARE"; *(s+1) = E; printf("%s\n", s);  union data {     char text[20];     int valoare; }u;  printf("%d\n", sizeof(u));  strcpy(u.text, "SARE"); u.valoare += 1&lt;&lt;10; printf("%s\n", u.text); return 0; }</pre>
---	---

3. (2p) Scrieți programul complet în C, care implementează funcțiile de mai jos. Citiți datele de intrare și afișați rezultatele în funcția main. Definim următoarea operație pe un număr natural x: șterge prima secvență de trei cifre consecutive egale care apare în număr. Implementați operația sub forma unei funcții care returnează numărul modificat. Dacă nu se poate aplica operația, se va returna numărul dat la intrare. Constrângeri:  $0 \leq x \leq 10^{15}$

Exemple: f(111222) returnează 222, f(123) returnează 123.

Un număr se cheamă triplicat dacă putem șterge toate cifrele lui aplicând operația de mai sus, de oricâte ori este necesar. Implementați o funcție care returnează 1 dacă numărul x dat la intrare este triplicat și 0 altfel.

Exemple: 122211 este triplicat, 797979 nu este.

4. (1p) Un schior înregistrează secvența de altitudini de pe traseul său sub forma unui tablou de numere naturale. Scrieți în C o funcție recursivă numită *vai* care returnează numărul total de vâi prin care a trecut schiorul. O vale este o valoare din interiorul tabloului (nu poate fi capăt!) care este strict mai mică decât ambele valori ale celor doi vecini ai săi (stânga, dreapta). Numele tabloului și numărul de elemente sunt trimise ca și argumente la apelul funcției. Constrângeri:  $0 < n \leq 100$ ,  $0 \leq a[i] < 10000$ .

Exemple:

Înainte de apel:	Apelul funcției:	Valoarea returnată:
$a \leftarrow [1500 \ 1500 \ 1400 \ 1560 \ 2000 \ 1900 \ 2500 \ 1800 \ 2000]$ $n \leftarrow 7$	$x \leftarrow \text{vai}(a, n)$	$x \rightarrow 3$
$a \leftarrow [2500 \ 2400 \ 2300 \ 2000 \ 1500]$ $n \leftarrow 5$	$x \leftarrow \text{vai}(a, n)$	$x \rightarrow 0$
$a \leftarrow [1500 \ 1300]$ $n \leftarrow 2$	$x \leftarrow \text{vai}(a, n)$	$x \rightarrow 0$
$a \leftarrow [1000]$ $n \leftarrow 1$	$x \leftarrow \text{vai}(a, n)$	$x \rightarrow 0$

5. (3p) Scrieți programul complet în C. Proiectați o structură pentru stocarea unei date calendaristice. Fișierul de intrare *input.txt* conține pe primul rând numărul  $n$  apoi pe următoarele  $n$  rânduri câte o dată calendaristică. Formatul pentru datele calendaristice poate fi de trei feluri:

- zz.ll.aaaa - stil românesc, câmpurile sunt separate prin punct;
- aaaa ll zz - stil maghiar, câmpurile sunt separate prin spațiu;
- ll. zz, aaaa - stil american, luna este dată de primele trei litere ale lunii în engleză, câmpurile sunt separate fie prin punct și spațiu fie prin virgulă și spațiu.

a) Citiți datele și stocați-le într-un șir de structuri alocate dinamic. Constrângeri:  $0 < n \leq 1000$ , datele sunt valide.

b) O dată calendaristică este repetitivă dacă numărul format din concatenarea câmpurilor se poate despărți în două părți identice (care conțin aceleași cifre și în aceeași ordine). Ordinea câmpurilor poate fi oricare, fiecare se completează cu zerouri pâna la două cifre și de la an considerăm doar ultimele două cifre. Determinați, folosind o funcție, care date calendaristice sunt repetitive dintre cele citite din fișier. Afișați *da* sau *nu* pentru fiecare caz.

<i>input.txt</i>	afișare ecran	explicații / exemple de concatenare și despărțire corectă
6 31.12.2023 2024 01 01 dec. 11, 1112 ian. 01, 1111 11.09.2011 11.11.1811	da nu nu nu nu da	luna zi an: 123   123  apare 0 în orice reordonare, de ex. 11110101 sau 01011111  111   111

c) Scrieți în fișierul text *output.txt* toate datele calendaristice valide și repetitive de la anul 1 până la anul 3000.