



1. Care dintre următoarele expresii C/C++ are valoarea 1 dacă și numai dacă variabila de tip *int* notată cu *x* reține o valoare pozitivă reprezentată prin exact trei cifre:

- a.  $x \geq 99 \ \&\& \ x \leq 999$
- b.  $x > 99 \ \&\& \ x \leq 999$
- c.  $x \geq 99 \ \&\& \ x < 999$
- d.  $x > 99 \ \&\& \ x < 999$

2. Un elev scrie un program care folosește metoda backtracking pentru a afișa toate modurile în care numărul 9 poate fi scris ca sumă de trei numere naturale nenule. Soluțiile afișate de acest program sunt, în ordine, următoarele:

7+1+1, 6+2+1, 5+3+1, 5+2+2, 4+4+1, 4+3+2, 3+3+3.

El rulează același program pentru a afișa modurile distincte în care poate fi scris numărul 10 ca sumă de trei numere naturale nenule. Care va fi a șasea soluție?

- a. 6+2+2
- b. 5+4+1
- c. 5+3+2
- d. 4+4+2

3. Un graf orientat cu 5 vârfuri, numerotate de la 1 la 5, are arcele:  
(1,4), (4,2), (2,5), (5,1), (1,2), (4,5), (2,1), (5,4).  
Indicați numărul vârfurilor pentru care gradul exterior este egal cu gradul interior.

- a. 2
- b. 3
- c. 4
- d. 5

4. Variabila *clasa* definită mai jos memorează numele și media generală pentru fiecare dintre cei 30 de elevi ai unei clase.

```
struct elev
{
    char nume[20];
    float medie;
} clasa[30];
```

Care dintre următoarele instrucțiuni afișează numele integral al unuia dintre elevii acestei clase:

- a. `cout << clasa.nume;`
- b. `cout << clasa.nume[0];`
- c. `cout << clasa[0].nume;`
- d. `cout << nume.clasa[0];`

5. Care dintre următoarele șiruri de valori poate constitui vectorul "de tați" pentru un arbore cu rădăcină care are 10 noduri numerotate de la 1 la 10 dintre care patru sunt de tip "nod frunză":

- a. 10, 6, 0, 3, 3, 5, 8, 3, 10, 4
- b. 10, 5, 0, 3, 3, 5, 8, 7, 10, 4
- c. 10, 5, 0, 3, 3, 5, 8, 3, 10, 4
- d. 10, 6, 0, 3, 3, 5, 8, 3, 1, 4

6. Se consideră algoritmul de mai jos, reprezentat în pseudocod. S-a notat cu  $a \% b$  restul împărțirii numărului natural  $a$  la numărul natural nenul  $b$ , cu  $[a]$  partea întreagă a numărului real  $a$ .

```
citește a, b, m ( $1 \leq a \leq b \leq 10000$ )
c ← 0
cat timp a ≤ b execută
    x ← a
    s ← 0
    cat timp x > 0 execută
        s ← s + x%10
        x ← [x/10]
    dacă s = m atunci
        c ← c+1
    a ← a+1
scrie c
```



**6a.** Scrieți în fereastra de concurs ce se afișează în urma executării algoritmului dacă se citesc valorile 8 30 9.

(6 puncte, 4 submit-uri)

**6b.** Dacă pentru  $a$  se citește valoarea 100 iar pentru  $m$  valoarea 6, scrieți în fereastra de concurs care este cea mai mică și cea mai mare valoare care se poate citi pentru  $b$  astfel încât în urma executării algoritmului, pentru fiecare dintre acestea să se afișeze valoarea 10?

(6 puncte, 4 submit-uri)

**6c.** Scrieți în fereastra de concurs programul C/C++ corespunzător algoritmului dat.

(10 puncte, 12 submit-uri)

**6d.** Scrieți **pe foaia de concurs** în pseudocod un algoritm echivalent cu cel dat, înlocuind adecvat prima structură cât timp cu o structură repetitivă cu număr cunoscut de pași.

(6 puncte)

**7.** Un graf neorientat are 9 noduri și următoarele proprietăți:

- are trei componente conexe;
  - gradul oricărui nod este un număr natural nenul.
- Scrieți în fereastra de concurs care este numărul minim și numărul maxim de muchii pe care le poate avea acest graf.

(6 puncte, 4 submit-uri)

**8.** Variabilele  $i$  și  $j$  sunt de tip întreg, iar variabila  $a$  memorează un tablou bidimensional cu 9 linii și 9 coloane, numerotate de la 0 la 8, având inițial toate elementele nule.

Scrieți în fereastra de concurs secvența de instrucțiuni de mai jos, înlocuind punctele de suspensie

```
for(i=0; i<=8; i++)
    for(j=0; j<=8; j++)
    {
        .....
    }
```

astfel încât, în urma executării secvenței obținute, variabila  $a$  să memoreze tabloul de mai jos:

```
2 2 2 2 2 2 2 1 1
2 2 2 2 2 2 1 1 1
2 2 2 2 2 1 1 1 2
2 2 2 2 1 1 1 2 2
2 2 2 1 1 1 2 2 2
2 2 1 1 1 2 2 2 2
2 1 1 1 2 2 2 2 2
1 1 1 2 2 2 2 2 2
1 1 2 2 2 2 2 2 2
```

ATENȚIE: Scrieți în fereastra de concurs strict instrucțiuni care atribuie valori elementelor matricei  $a$ . Matricea  $a$  și variabilele  $i$  și  $j$  sunt declarate în prealabil. NU le declarați deoarece în acest caz programul va lua 0 puncte (erori de compilare)! NU scrieți în fereastra de concurs instrucțiuni de citire sau afișare.

(6 puncte, 4 submit-uri)

**9.** Un număr natural  $n$  este perfect dacă este egal cu suma divizorilor săi strict mai mici decât el. De exemplu, 6 este număr perfect deoarece are ca divizori mai mici ca el numerele 1, 2 și 3 iar  $6=1+2+3$ . Euler a demonstrat că un număr par este perfect dacă și numai dacă este de forma  $2^{n-1} \cdot (2^n - 1)$  cu  $n$  număr natural iar  $2^n - 1$  este număr prim. De exemplu, pentru  $n = 2$  această expresie devine  $2^{2-1} \cdot (2^2 - 1) = 2 \cdot 3 = 6$  (observăm că 3 este număr prim). Pornind de la această formulă s-au determinat numere perfecte foarte mari, cel mai mare dintre acestea având 82.048.640 cifre.

Funcția  $p$  are un singur parametru  $n$  număr natural,  $2 \leq n \leq 20$ . Funcția returnează valoarea 1



dacă numărul egal cu expresia  $2^{n-1} \cdot (2^n - 1)$  este perfect, și 0 în caz contrar. Scrieți în fereastra de concurs a răspunsului definiția completă a funcției  $p$ .

Exemple:

- dacă  $n=3$ , funcția returnează valoarea 1
- dacă  $n=4$ , funcția returnează valoarea 0 (deoarece  $2^{4-1} \cdot (2^4 - 1) = 8 \cdot 15 = 120$  NU este număr perfect deoarece 15 nu este număr prim)

ATENȚIE: Scrieți în fereastra de concurs definiția funcției cerute. Dacă doriți, puteți defini și alte funcții pe care funcția cerută le apelează, caz în care scrieți în fereastra de concurs și definițiile acestor funcții. Dacă rezolvarea dumneavoastră folosește elemente aflate în diverse biblioteci, altele în afară de *iostream* scrieți în fereastra de concurs și declarațiile de includere pentru acestea. NU scrieți în fereastra de concurs funcția `main()` !

(10 puncte, 12 submit-uri)

**10.** Fiind dat un număr natural  $n$  ( $1 \leq n \leq 26$ ), numim  $n$ -*cuvânt* un cuvânt care conține  $n$  litere mici distincte. De exemplu, fiecare dintre cuvintele "an", "ana" și "anna" sunt 2-*cuvânt* iar "abracadabra" este 5-*cuvânt*.

Scrieți în fereastra de concurs un program care citește de la tastatură în această ordine: un număr natural  $m$  ( $1 \leq m \leq 30$ ),  $m$  cuvinte și un număr natural  $n$  ( $1 \leq n \leq 26$ ). Fiecare cuvânt citit are cel mult 40 de caractere, este format numai din litere mici ale alfabetului englez și după fiecare cuvânt se apasă tasta *Enter*. Programul determină cel mai lung cuvânt dintre cele  $m$  citite cu proprietatea că este  $n$ -*cuvânt* și afișează pe ecran lungimea acestuia. Dacă niciunul dintre cele  $m$  cuvinte citite nu este  $n$ -*cuvânt* programul afișează mesajul *nu exista*.

Exemplu:

Dacă se citesc următoarele

7

abc

ana

avea

clementine

ananas

mere

piersici

3

programul afișează 6

Explicații: dintre cele 7 cuvinte "abc", "avea", "ananas" și "mere" sunt 3-*cuvinte*. Cel mai lung dintre acestea este "ananas" care are 6 caractere.

(10 puncte, 12 submit-uri)

**11.** Pentru România, Administrația Națională a Vremii definește o perioadă caniculară situația când pe parcursul a cel puțin trei zile consecutive temperatura maximă depășește o valoare denumită prag diurn (valoare notată  $PD$ ), iar temperatura minimă nocturnă nu scade sub o valoare denumită prag nocturn ( $PN$ ). Aceste praguri pot varia în funcție de regiunea țării și de particularitățile climatice locale. De exemplu, pentru localitățile aflate în zonele de câmpie,  $PN = 20^\circ$  iar  $PD = 35^\circ$ .

Elevii unui liceu au alcătuit o listă cu temperaturile minime și maxime măsurate într-un interval de observație format din zile consecutive. Valorile au fost reținute în ordinea în care au fost măsurate: prima valoare din listă este temperatura minimă din prima zi a intervalului, a doua valoare este temperatura maximă din prima zi, a treia valoare reprezintă temperatura minimă din a doua zi, a patra valoare este temperatura maximă din a doua zi ș.a.m.d.

Fișierul *meteo.in* conține cel puțin 8 și cel mult  $3 \cdot 10^6$  valori numerice întregi fiecare dintre acestea aparținând intervalului  $[-100, 100]$ . Pe prima linie sunt două valori  $PN$  și  $PD$  în această ordine, separate printr-un spațiu. A doua linie a fișierului conține un număr par de valori separate prin câte



un spațiu. Aceste valori reprezintă lista de temperaturi alcătuită de elevi. Se cere să se afișeze pe ecran, separate prin exact un spațiu, două valori: prima reprezintă numărul de perioade de caniculă din intervalul de observație iar a doua reprezintă durata exprimată în zile a celei mai lungi perioade de caniculă. Dacă în respectivul interval nu a fost nicio perioadă de caniculă pe ecran se va afișa mesajul *nu exista*. Proiectați un algoritm eficient din punctul de vedere al spațiului de memorie utilizat și al timpului de executare.

Exemplu: dacă fișierul *meteo.in* are următorul conținut

20 35

21 30 19 35 **25 36 25 37 26 40 22 36 19**

36 22 35 22 36 20 33 **21 36 20 36 21 37**

se afișează pe ecran valoarea **2 4**

Explicații: O zi poate face parte dintr-o perioadă caniculară dacă temperatura minimă din acea zi este mai mare sau egală ca 20 iar temperatura maximă este strict mai mare ca 35. În intervalul de observație care a durat 13 zile, există două perioade caniculare: prima începe în a treia zi a intervalului și durează patru zile; a doua începe în a unsprezecea zi a intervalului și durează trei zile.

**11a.** Descrieți **pe foaia de concurs** în limbaj natural algoritmul utilizat, justificând eficiența acestuia.

(2 puncte)

**11b.** Scrieți în fereastra de concurs programul C/C++ corespunzător algoritmului descris.

(8 puncte, 12 submit-uri)