

SUBIECTUL al II-lea

(40 de puncte)

- Algoritmul alăturat este reprezentat în pseudocod.
S-a notat cu $a \div b$ restul împărțirii numărului natural a la numărul natural nenul b și cu $[c]$ partea întreagă a numărului real c .
 - Scieți valoarea afișată în urma executării algoritmului dacă se citește numărul 412531. (6p.)
 - Scieți două numere din intervalul $[1, 11110]$ care pot fi citite pentru variabila n , astfel încât, pentru fiecare dintre acestea, în urma executării algoritmului, să se afișeze valoarea 11111. (6p.)
 - Scieți programul C/C++ corespunzător algoritmului dat. (10p.)
 - Scieți în pseudocod un algoritm echivalent cu cel dat, înlocuind adecvat structura `cât timp...execută` cu o structură repetitivă cu test final. (6p.)
- Într-un tablou unidimensional cu 7 elemente, trei dintre acestea au valorile 2, 10 respectiv 24. Pentru a verifica dacă în tablou există elementul cu valoarea $x=16$, se aplică metoda căutării binare. Scieți un exemplu de valori naturale pe care le pot avea elementele tabloului, în ordinea în care acestea apar în tablou, astfel încât x să se compare cu exact două elemente pe parcursul aplicării metodei indicate. (6p.)
- Gama muzicală conține, în această ordine, notele **do, re, mi, fa, sol, la, si**. Două note formează un interval muzical, iar acesta este numit **terță** dacă, în ordinea din gamă, între cele două note există o singură altă notă muzicală. În notația englezescă se folosesc unele litere mari ale alfabetului pentru notele muzicale, astfel: litera **A** pentru nota **la**, litera **B** pentru nota **si**, apoi, în ordine, litera **C** pentru nota **do**, litera **D** pentru nota **re**, litera **E** pentru nota **mi**, litera **F** pentru nota **fa** și litera **G** pentru nota **sol**. Variabilele **nota1** și **nota2** memorează literele corespunzătoare notației englezești pentru două note care formează un interval muzical, date în ordinea din gamă, **nota1** memorând una dintre literele **C, D, E, F** sau **G**.
Declarați variabilele **nota1** și **nota2** și scieți o secvență de instrucțiuni C/C++ în urma executării căreia să se afișeze pe ecran mesajul **TERTA**, dacă intervalul respectiv formează o terță, sau mesajul **NU**, în caz contrar.
Exemplu: dacă în variabila **nota1** se memorează litera **G** (pentru nota **sol**), iar în variabila **nota2** se memorează litera **B** (pentru nota **si**), se afișează mesajul **TERTA**. (6p.)

SUBIECTUL al III-lea

(30 de puncte)

- Un număr natural nenul, n , se numește **moderat** dacă este egal cu produsul a două numere prime, iar acestea sunt consecutive în șirul numerelor prime (2, 3, 5, 7, 11, 13, 17...).
Se citește un număr natural nenul, n , și se cere să se scrie valoarea 1, dacă n este un număr moderat, sau valoarea 0, în caz contrar. Scieți în pseudocod algoritmul corespunzător.
Exemplu: dacă $n=35$, se scrie 1 ($35=5 \cdot 7$), iar dacă $n=18$ sau $n=55$ sau $n=4$, se scrie 0. (10p.)
- O grădină este împărțită în parcele pătrate identice, dispuse succesiv, pe un rând. O parcelă conține același tip de plante (doar flori sau doar gazon) și oricare două parcele alăturate conțin tipuri diferite de plante, **pe prima parcelă fiind gazon**. Pentru fiecare parcelă se memorează înălțimea acesteia, egală cu înălțimea medie a plantelor conținute, exprimată în decimetri. Pentru a pune în valoare florile, orice parcelă cu flori trebuie să fie mai înaltă cu cel puțin un decimetru decât parcela cu gazon vecină anterioară. S-a hotărât tunderea parcelor cu gazon prea înalte, astfel încât ele să fie aduse la înălțimea maximă permisă.
Scieți un program C/C++ care citește de la tastatură un număr natural din intervalul $[2, 10^2]$, n , apoi cele n elemente ale unui tablou unidimensional, numere naturale din intervalul $[2, 10]$, reprezentând înălțimile parcelor, în ordinea dispunerii acestora în grădină. Programul modifică apoi tabloul în memorie corespunzător situației terenului după tunderea gazonului și afișează pe ecran tabloul obținut, cu elementele separate prin câte un spațiu.
Exemplu:
pentru $n=6$ și tabloul

6	2	3	7	9	4
---	---	---	---	---	---

 se obține tabloul

1	2	3	7	3	4
---	---	---	---	---	---

 (10p.)
- Șirul 2, 3, 5, 9, 17, 33, 65, 129 . . . este definit astfel: $f_1=2$, $f_2=3$, $f_n=3 \cdot f_{n-1}-2 \cdot f_{n-2}$ (unde n este un număr natural $n \geq 3$).
Se citesc de la tastatură două numere naturale x și y ($x < y \leq 10^9$), valorile a doi termeni aflați pe poziții consecutive în șirul dat, și se cere să se scrie în fișierul **bac.txt**, în ordine strict descrescătoare, separați prin câte un spațiu, toți termenii șirului care sunt mai mici sau egali cu y . Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.
Exemplu: dacă se citesc numerele 33 65, fișierul conține numerele 65 33 17 9 5 3 2
 - Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)
 - Scieți programul C/C++ corespunzător algoritmului proiectat. (8p.)