

EXAMEN ADMITERE INFORMATICĂ – iulie 2013

Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu. Timpul efectiv de lucru este de 3 ore.

Subiectul I (30 puncte)

- Ce înțelegeți prin *complexitate* a unui algoritmul? Dați câte un exemplu sugestiv de algoritmul (împreună cu complexitatea lui) pentru *complexitate ca durată de executare* (timp) și *complexitate ca spațiu de memorie utilizat*.
- Definiți noțiunea de subprogram de tip funcție. Dați un exemplu sugestiv de definiție și apel într-un limbaj de programare.
- Definiți șirul lui Fibonacci și scrieți un algoritmul pentru determinarea elementului de pe poziția  $k$  din șirul lui Fibonacci. Nu se vor folosi tablouri pentru memorarea elementelor șirului.

Subiectul II (30 puncte)

Se dă următorul algoritmul:

```
Citeste n;  
Citeste v;  
Pentru i ← 1, n-1 executa  
  d ← v;  
  Citeste a;  
  Daca a ≠ 0 atunci  
    gasit ← fals;  
    Cattimp (d ≤ v · a) si (¬ gasit) executa  
      Daca ((d/a) · a = d) si ((d/v) · v = d) atunci  
        gasit ← adevarat  
      altfel  
        d ← d+1;  
    SfDaca;  
  SfCattimp;  
SfDaca;  
v ← d;  
SfPentru;  
Tipareste v;
```

Se cere:

- Ce se va afișa dacă se citesc valorile:  
5, 8, 24, 16, 64, 192?
- Determinați un set de date de intrare nenule care să înceapă cu valoarea 4 astfel încât valoarea afișată să fie egală cu 1002.
- Determinați un set de date de intrare nenule care să înceapă cu valorile 3 și 25 astfel încât valoarea afișată să fie egală cu 225.

**Observație.** Prin „¬” s-a notat operatorul logic NOT (negația logică).

Subiectul III (30 puncte)

Se citește un șir  $X$  de  $n$  numere naturale pozitive, citirea șirului terminându-se la introducerea valorii 0. (Exemplu: dacă valorile introduse sunt 1, 2, 3, 0 atunci șirul citit va fi  $X = (x_1 = 1, x_2 = 2, x_3 = 3)$ , iar lungimea șirului citit va fi  $n = 3$ ), unde  $1 \leq n \leq 500$ ,  $1 \leq x_i \leq 10000$ . Să se scrie un program care construiește și afișează șirul  $Y = ((y_1, f_1), (y_2, f_2), \dots, (y_k, f_k))$  unde

- $y_1, y_2, \dots, y_k$  reprezintă, în ordine crescătoare, numerele distincte din șirul  $X$  cu proprietatea că suma cifrelor fiecărui număr  $y_i$  ( $\forall i, 1 \leq i \leq k$ ) este un număr prim.
- $f_i$  ( $\forall i, 1 \leq i \leq k$ ) reprezintă numărul de apariții a valorii  $y_i$  în șirul  $X$ .

Șirul  $Y$  se va construi direct ordonat (după valorile  $y_i$ ), fără a se face ordonarea după construcție.

Se vor scrie:

- Subprogram pentru citirea unui șir.
- Subprogram pentru verificarea dacă un număr este prim.
- Subprogram pentru determinarea sumei cifrelor unui număr.
- Subprogram pentru inserarea în șirul  $Y$  a unei valori cu proprietatea cerută.
- Subprogram pentru construirea șirului  $Y$ .
- Subprogram pentru tipărirea unui șir.
- Programul principal.

**Exemple:**

- Pentru șirul  $X = (142, 13, 89, 21, 91, 11, 8, 142, 21)$  se obține  $Y = ((11, 1), (21, 2), (89, 1), (142, 2))$ .
- Pentru șirul  $X = (6, 15, 103)$  se va tipări mesajul '**Sirul Y este vid**'.

Programul se poate scrie într-unul dintre limbajele studiate la liceu (Pascal, C++ etc). Folosiți comentarii pentru a ușura înțelegerea soluției date (explicarea semnificației identificatorilor folosiți, descrierea detaliilor de implementare etc).

**UNIVERSITATEA BABEȘ-BOLYAI CLUJ-NAPOCA**  
**FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ**

**EXAMEN ADMITERE – iulie 2013**  
**INFORMATICĂ**

**BAREM**  
Corectare INFORMATICĂ

**SUBIECT I**

- a) **8p** din care
- complexitate *durata de executare* **2p**
  - complexitate *spațiu de memorare* **2p**
  - fiecare exemplu **2p**
- b) **10p** din care
- noțiunea de subprogram de tip funcție **5p**
  - exemplu definire **3p**
  - exemplu apel **2p**
- c) **12p** din care
- definire șir Fibonacci **3p**
  - algoritm **9p**
- (pentru soluție cu tablou se acordă maxim 4p)

**SUBIECT II**

- a) Se afișează valoarea 192. **4p**  
justificare **4p**
- b) 4 1002 1002 1002 1002 **6p**  
justificare **5p**
- c) 3 25 9 15 **6p**  
justificare **5p**

**SUBIECT III**

- a). Subprogram pentru citirea unui șir **3p**
- b). Subprogram pentru verificarea dacă un număr este prim **4p**
- c). Subprogram pentru determinarea sumei cifrelor unui număr **4p**
- d). Subprogram pentru inserarea în șirul  $Y$  a unei valori cu proprietatea cerută **6p**
- e). Subprogram pentru construirea șirului  $Y$  **3p**
- f). Subprogram pentru tipărirea unui șir **3p**
- g). Program principal **2p**
- Stil **5p**
- comentarii, structurare, indentare, folosirea subprogramelor, apelul corect al subprogramelor, comunicarea între subprograme și programul apelant prin parametri.

**Comisia de admitere**