





2. Algoritmul alăturat este reprezentat în pseudocod.

S-a notat cu  $a \% b$  restul împărțirii numărului natural  $a$  la numărul natural nenul  $b$  și cu  $[a]$  partea întreagă a numărului real  $a$ .

- Scieți valoarea afișată dacă se citește șirul de valori 5, 8, 50, 3, 40, 60, 100. (6p)
- Pentru  $n=3$  și  $a=4$ , scieți două seturi diferite de valori distincte care pot fi citite pentru variabila  $b$  astfel încât, în urma executării algoritmului, valoarea afișată să fie mai mică sau egală cu 20. (două seturi sunt diferite dacă diferă prin cel puțin o valoare) (6p)
- Scieți programul C/C++ corespunzător algoritmului dat. (10p)
- Scieți în pseudocod un algoritm, echivalent cu cel dat, care să utilizeze o singură structură repetitivă. (6p)

```

citește n, a
(numere naturale)
s ← 0
pentru i ← 1, n execută
    citește b (număr natural)
    x ← a
    t ← 0
    cât timp b ≥ 1 execută
        dacă b % 2 = 0
            atunci
                x ← x * 2
                b ← [b / 2]
            altfel
                t ← t + x
                b ← b - 1
    s ← s + t
scrie s
    
```

- Variabilele  $i$  și  $j$  sunt de tip întreg, iar variabila  $a$  memorează un tablou bidimensional cu 6 linii și 6 coloane, numerotate începând de la 0, cu elemente numere întregi, iar inițial toate au valoarea -1. Fără a utiliza alte variabile decât cele menționate, scieți o secvență de instrucțiuni astfel încât, în urma executării acesteia, variabila  $a$  să memoreze tabloul alăturat. (6p.)

```

5 4 3 2 1 0
1 2 3 4 0 1
5 4 3 0 4 2
1 2 0 3 3 3
5 0 2 4 2 4
0 5 1 5 1 5
    
```

**SUBIECTUL al III-lea**

**(30 de puncte)**

- Scieți definiția completă a subprogramului C++ `numar` care are 3 parametri:  $a$ , prin care primește un tablou unidimensional cu maximum 100 de numere naturale mai mici decât 100000;  $n$ , numărul efectiv de elemente ale tabloului ( $0 < n \leq 100$ ),  $k$ , un număr natural. Subprogramul furnizează prin intermediul parametrului de ieșire  $k$ , poziția ultimului număr din vector, care are număr maxim de divizori primi. Exemplu: Dacă  $n=8$ ,  $a=(9, 6, 30, 15, 70, 42, 8, 10)$ , obținem  $k=5$ , deoarece 42 este ultimul număr care are număr maxim de divizori primi. (10p.)

- Se dă un șir de maximum 255 de caractere în care cuvintele sunt formate din litere mici ale alfabetului englez și sunt separate prin spații și virgule. Un cuvânt este palindrom dacă se citește la fel în ambele direcții. De exemplu, cuvântul `capac` este palindrom. Să se afișeze cuvintele palindrom distincte din text, în ordinea descrescătoare a numărului de vocale conținute, fiecare cuvânt fiind afișat pe câte o linie. Dacă două cuvinte palindrom au același număr de vocale atunci ele se vor ordona lexicografic.

Exemplu:

Dacă se citește șirul

`ana am stricat aerisirea cu un cojoc cand ma urcam in dud urmarind un cuc rar care era foarte sus`

se afișează pe ecran:

aerisirea

ana

cojoc

cuc

dud

rar

sus

(10p.)

- Cifra de control** a unui număr se obține efectuând suma cifrelor sale, apoi suma cifrelor acestei sume, până când suma obținută este un număr format dintr-o singură cifră. Această ultimă cifră poartă numele de **cifră de control**. Exemplu: Cifra de control ( $cc$ ) a numărului  $n=4568258$  este  $cc(4568258) = cc(38) = cc(11) = 2$ .

Se citesc din fișierul `bac.txt` cel mult 1000000 numere naturale cu maxim 9 cifre fiecare, până la apariția lui 0 care nu face parte din șir. Folosind un algoritm eficient din punct de vedere al memoriei și al timpului de execuție, să se determine și să se afișeze pe ecran numărul perechilor de elemente  $(a_i, a_j)$ , cu  $i \neq j$ , reprezentând pozițiile elementelor din șir, pentru care cifra de control a elementului  $a_i$  este egală cu cifra de control a elementului  $a_j$ .

Exemplu: Pentru fișierul `bac.txt` cu următorul conținut: `5678 12 1020 6 78 40803 11001 33 0`, se va afișa pe ecran valoarea 9.

- Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)

- Scieți programul C/C++ corespunzător algoritmului proiectat. (8p.)