Adrian Runceanu

Laborator nr. 2
Algoritmi elementari (pseudocod)
Structuri repetitive - Structura repetitivă cu test iniţial

A. Probleme rezolvate:

1) Descompunerea în factori primi ai unui număr a. Se citeşte un număr întreg a. Să se realizeze un algoritm care să afişeze factorii primi şi puterile lor pentru numărul citit.

Solutie:

Pas 1. Datele de intrare: a numar intreg

Pas 2. Analiza problemei: Algoritmul urmăreşte pas cu pas procedeul matematic de descompunere în factori primi.
Se folosesc două structuri repetitive cu test iniţial, imbricate.
Prima structură asigură repetarea instrucţiunilor cât timp numărul nu a ajuns la 1.
A doua structură numără în variabila p (care se iniţializează cu 0 pentru fiecare nouă valoare a lui d) de câte ori se poate face împarţirea numărului la divizorul d. În cazul în care s-a putut împarţi a la d (deci p este diferit de 0), se afişează divizorul şi puterea lui.
Apoi se creşte d şi se repetă instrucţiunile pentru a se verifica dacă acest nou număr este divizor şi a afla puterea.

Pas 3. Scrierea algoritmului în pseudocod:

```
întreg a, d, p
citeşte a
d ← 2 // primul factor prim este 2
cât timp a > 1 execută
|    p ← 0 // puterea factorului d este 0
|    cât timp (a % d = 0) execută
|    |    p ← p + 1
|    |    a ← a / d
|    |■
|    dacă p ≠ 0 atunci
|    |    scrie d, „^”, p, „,”
|    |■
|    d ← d + 1
|■
stop
```

De exemplu:
Dacă se citeşte pentru **a** valoarea 36 atunci algoritmul va afişa „**2^2, 3^2**”.

36	2
18	2
9	3
3	3
1	

2) Cel mai mare divizor comun intre 2 numere întregi a şi b. Se citesc două numere întregi a şi b. Să se realizeze un algoritm care să afişeze cmmdc(a,b).

Solutie:

Pas 1. Datele de intrare:

Pas 2. Analiza problemei: Algoritmul folosit este **algoritmul lui EUCLID**.

Exista şi algoritmul care afla cmmdc prin scadere însa nu este eficient.
Algoritmul lui Euclid foloseşte o structură repetitivă cu test iniţial.
Mai întâi aflăm restul împărţirii lui a la b şi cât timp acest rest este diferit de 0, vom înlocui pe a cu b şi pe b cu restul obţinut, după care recalculăm restul împărţirii noului a la noul b.
Euclid a demonstrat că oricare ar fi numerele a şi b iniţiale, repetând operaţiile descrise mai sus, găsim restul = 0. În acel moment putem afirma că cmmdc(a,b) este ultimul rest nenul. Deoarece variabila b ia mereu valoarea restului, afişam pe b ca fiind cmmdc.

Pas 3. Scrierea algoritmului în pseudocod:

```
întreg a, b, r
citeşte a, b
r ← a % b // r reţine restul împărţirii lui a la b
cât timp r ≠ 0 execută
|     a ← b
|     b ← r
|     r ← a % b
|■
scrie "cmmdc = ", b
stop
```

De exemplu:
Dacă se citeşte a = 36 şi pentru b = 24 atunci algoritmul va afişa **cmmdc = 12**

3) Numărul invers. Se citeşte un număr întreg a. Să se realizeze un algoritm care să afiseze numărul invers. Numim *număr invers* (sau **oglindit**) numărul format cu cifrele citite de la dreapta la stanga.

Solutie:

Pas 1. Datele de intrare: a numar intreg

Pas 2. Analiza problemei: Algoritmul descompune în cifre numărul a şi adaugă fiecare cifră la numărul invers.
Pas 3. Scrierea algoritmului în pseudocod:

```
întreg a, inv
citeşte a
inv ← 0 // initial valoarea in variabila inv este zero
cât timp a ≠ 0 execută
|     inv ← inv * 10 + a % 10
|     a ← a / 10
|■
scrie 'Numarul invers este  = ', inv
stop
```

De exemplu:
Dacă se citeşte a = 327 atunci algoritmul va afişa **Numarul invers este = 723**

4) Numărul palindrom. Se citeşte un număr întreg a. Să se realizeze un algoritm care să verifice dacă numărul citit este sau nu palindrom. Numim **palindrom** *un număr care este egal cu oglinditul său*.

Solutie:

Pas 1. Datele de intrare: a numar intreg

Pas 2. Analiza problemei: Algoritmul se bazează pe problema anterioară, de calcul al numărului invers. *Un număr este palindrom dacă numărul iniţial citit este egal cu inversul sau.*
De aceea algoritmul descompune în cifre numărul a şi formează numărul invers, dupa care compară acest număr invers cu numărul iniţial citit.
Am descompus în cifre o copie a numărului „a" deoarece în structura repetitivă se modifică valoarea numărului introdus, iar noi avem nevoie de valoarea iniţiala pentru verificare.

Pas 3. Scrierea algoritmului în pseudocod:

```
întreg a, inv, aux
citeşte a
aux ← a // copiez valoarea iniţială a lui a în aux
inv ← 0 // iniţial inv are valoarea zero
cât timp aux ≠ 0 execută
|     inv ← inv * 10 + aux % 10
|     aux ← aux / 10
|■
dacă inv = a atunci
|     scrie „numarul este PALINDROM”
|altfel
|     scrie „numarul NU este PALINDROM”
|■
stop
```

De exemplu:
Dacă se citeşte pentru **a** valoarea 323 atunci algoritmul va afişa „**numarul este PALINDROM**”, iar dacă va citi 123 va afişa „**numarul NU este PALINDROM**”.

5) Suma cifrelor unui numar. Se citeşte un număr întreg n. Să se realizeze un algoritm care să calculeze suma cifrelor numărului dat.

Solutie:

Pas 1. Datele de intrare: n număr întreg

Pas 2. Analiza problemei:
Se împarte numărul la baza 10 si se adună restul împărtirii la 10 care reprezintă de fiecare dată ultima cifră a numărului. Numărul se modifică prin înlocuirea sa cu catul împărtirii lui la 10. Structura repetitivă se finalizează cand numărul devine 0 (zero).

Pas 3. Scrierea algoritmului în pseudocod:

```
întreg n, suma, rest, cat
citeste n
suma ← 0
cât timp n ≠ 0 execută
|     rest ← n % 10
|     cat ← n / 10
|     suma ← suma + rest
|     n ← cat
|■
scrie ‘Suma cifrelor = ’, suma
stop
```

De exemplu:
Dacă se citeşte n = 12345 atunci algoritmul va afişa **Suma cifrelor = 15**
Iar dacă va citi n = 789 va afişa **Suma cifrelor = 24**

6) Aflarea primei cifre a numărului natural n. Se citeşte un număr întreg n. Să se realizeze un algoritm care să afişeze prima cifră a numărului dat.

Solutie:

Pas 1. Datele de intrare: n număr întreg

Pas 2. Analiza problemei:
Se împarte numărul la baza 10. Numărul se modifică prin înlocuirea sa cu catul împărtirii lui la 10. Structura repetitivă se finalizează cand numărul devine mai mic strict decat 10 (zece).

Pas 3. Scrierea algoritmului în pseudocod:

```
întreg n, suma, rest, cat
citeste n
suma ← 0
cât timp n >9 execută
|     cat ← n / 10
|     n ← cat
|■
scrie 'Prima cifra = ',n
stop
```

De exemplu:
Dacă se citeşte n = 7234 atunci algoritmul va afişa **Prima cifra = 7**
Iar dacă va citi n = 1789 va afişa **Prima cifra = 1**

7) Aflarea cifrei maxime dintr-un număr n. Se citeşte un număr întreg n. Să se realizeze un algoritm care să afişeze cifra maximă a numărului dat.

Solutie:

Pas 1. Datele de intrare: n număr întreg

Pas 2. Analiza problemei:
Se împarte numărul la baza 10. Numărul se modifică prin înlocuirea sa cu catul împărtirii lui la 10. Fiecare cifra a numarului se compara cu o variabila in care se retine cifra maxima. Initial, aceasta variabla are valoarea 0 (zero). Structura repetitivă se finalizează cand numărul devine 0 (zero).

Pas 3. Scrierea algoritmului în pseudocod:

```
întreg n, suma, ultima_cifra, cifra_maxima
citeste n
cifra_maxima ← 0
suma ← 0
cât timp n ≠ 0 execută
|       ultima_cifra ← n % 10
|       dacă ultima_cifra > cifra_maxima atunci
|       |       cifra_maxima ← ultima_cifra
|       |■
|       n ← n / 10
|■
scrie 'Cifra maxima = ', cifra_maxima
stop
```

De exemplu:
Dacă se citeşte n = 7284 atunci algoritmul va afişa **Cifra maxima = 8**,
Iar dacă va citi n = 9789 va afişa **Cifra maxima = 9**

8) Aflarea cifrei minime dintr-un număr n. Se citeşte un număr întreg n. Să se realizeze un algoritm care să afişeze cifra minimă a numărului dat.

Solutie:

Pas 1. Datele de intrare: n număr întreg

Pas 2. Analiza problemei:
Se împarte numărul la baza 10. Numărul se modifică prin înlocuirea sa cu catul împărtirii lui la 10. Fiecare cifra a numarului se compara cu o variabila in care se retine cifra minima. Initial, aceasta variabla are valoarea 10 (zece). Structura repetitivă se finalizează cand numărul devine 0 (zero).

Pas 3. Scrierea algoritmului în pseudocod:

```
întreg n, suma, ultima_cifra, cifra_minima
citeste n
cifra_minima ← 10
suma ← 0
cât timp n ≠ 0 execută
|       ultima_cifra ← n % 10
|       dacă ultima_cifra < cifra_minima atunci
|       |       cifra_minima ← ultima_cifra
|       |■
|       n ← n / 10
|■
scrie 'Cifra minima = ', cifra_minima
stop
```

De exemplu:
Dacă se citeşte n = 7284 atunci algoritmul va afişa **Cifra minima = 2**
Iar dacă va citi n = 9781 va afişa **Cifra minima = 1**

9) Numărul de cifre pare dintr-un număr n. Se citeşte un număr întreg n. Să se realizeze un algoritm care să afişeze câte cifre pare are numărul dat.

Solutie:

Pas 1. Datele de intrare: n număr întreg

Pas 2. Analiza problemei:
Se împarte numărul la baza 10. Numărul se modifică prin înlocuirea sa cu catul împărțirii lui la 10. Pentru fiecare cifra a numărului se verifica restul împărțirii la 2 dacă este zero (adica daca cifra este para). Dacă este îndeplinită condiția atunci se mareste cu o unitate o variabila care reține numărul de cifre pare. Initial, aceasta variabla are valoarea 0 (zero). Structura repetitivă se finalizează cand numărul devine 0 (zero).

Pas 3. Scrierea algoritmului în pseudocod:

```
întreg n, cifra, numar_cifre_pare
citeste n
numar_cifre_pare ← 0
cât timp n ≠ 0 execută
|     cifra ← n % 10
|     dacă cifra % 2 = 0 atunci
|     |     numar_cifre_pare ← numar_cifre_pare + 1
|     |■
|     n ← n / 10
|■
scrie 'Numarul de cifre pare = ', numar_cifre_pare
stop
```

De exemplu:
Dacă n = 5428, atunci se afiseaza '**Numarul de cifre pare = 3**' (4, 2 și 8)
iar dacă n = 49285640, atunci se afiseaza '**Numarul de cifre pare = 6**' (4, 2, 8, 6, 4, 0)

B. Probleme propuse spre rezolvare:

L2.1) *Aflarea ultimei cifre a numărului natural n.* Se citeşte un număr întreg n. Să se realizeze un algoritm care să afişeze ultima cifră a numărului dat.
Exemplu:
Pentru n = 12345 se va afişa valoarea 5
Iar pentru n = 789 se va afişa valoarea 9

L2.2) *Cifre distincte.* Se citeşte un număr întreg n. Să se realizeze un algoritm care să spună dacă un număr are toate cifrele distincte.
Exemplu:
Daca n = 545471236, atunci se va afisa mesajul **"Numarul NU are toate cifrele distincte"** (deoarece cifrele 5 și 4 apar de cate două ori în număr)
Daca n = 4321, atunci se va afisa mesajul **"Numarul are toate cifrele distincte"** (deoarece nici o cifră a sa nu se repeat)

L2.3) *Ecuația de gradul 1.* Fie ecuația a * x = b cu a si b numere reale. Să se realizeze un algoritm care să calculeze x.
Precizare: Ecuația poate avea multiple soluții, o soluţie sau niciuna!
Exemplu:
Daca avem valorile a = 2 si b = 3, atunci x = 3/2 = 1.5
Daca avem valorile a = 0 si b = 7, atunci x nu are valori
Daca avem valorile a = 0 si b = 0, atunci x poate lua orice valori

L2.4) *Cel mai mare divizor propriu al unui numar.* Se citește un număr n. Să se realizeze un algoritm care să afișeze cel mai mare divizor propriu al lui n (strict mai mic decât n).
Exemplu:
Dacă n = 24 cel mai mare divizor propriu este 12.
Dacă n = 7 cel mai mare divizor propriu este 1.
Dacă n = 125 cel mai mare divizor propriu este 25.
Dacă n = 175 cel mai mare divizor propriu este 35.

L2.5) *Numar prim.* Se citește un număr n. . Să se realizeze un algoritm care să spună dacă n este prim. Un număr este prim dacă nu se împarte decât la 1 și la el însuși.
Exemplu:
Dacă n = 12 atunci se va afisa mesajul "Numarul NU este prim"
Dacă n = 7 atunci se va afisa mesajul "Numarul este prim"

L2.6) *Numere prime pana la n.* Se citește un număr n. Să se realizeze un algoritm care să afișeze toate numerele prime mai mici sau egale cu n.
Exemplu:
Dacă n = 11 atunci se va afisa 2, 3, 5, 7, 11
Dacă n = 20 atunci se va afisa 2, 3, 5, 7, 11, 13, 17, 19

L2.7) https://www.pbinfo.ro/probleme/3166/vas1
Într-un vas sunt x litri de apă (x>0). După fiecare t minute, x/i din cantitatea de apă rămasă se evaporă. Să se realizeze un algoritm care să determine după câte minute vor rămâne în vas cel mult y litri de apă.
Exemplu:
Daca se dau valorile 100 15 20 5, atunci si va afisa 120
Explicație
x=100, t=15, y=20, i=5. Sunt necesare T=120 minute pentru ca în vas să rămână cel mult y=20 litri de apă.

L2.8) https://www.pbinfo.ro/probleme/1650/acelasinumar
Se dă un număr întreg n și alte k numere întregi. Să se realizeze un algoritm care să afle dacă, adunând toate cele k numere la n se obține o valoare egală cu valoarea inițială a lui n.
Exemplu:
Daca se dau valorile n = 25, k = 3 si urmatoarele n valori: 3, 16, -9, 3, atunci se va afisa mesajul NU
Explicație
25 + 16 - 9 + 3 = 35, număr diferit de cel inițial (25).

L2.9) https://www.pbinfo.ro/probleme/2601/sumapatratecifre
Să se realizeze un algoritm care să afișeze suma patratelor cifrelor unui număr natural de trei cifre citit de la tastatură.
Exemplu:
Intrare
Dacă n = 123 atunci se va afisa 14 (1^2+2^2+3^2 = 1 + 4 + 9 = 14)

L2.10) https://www.pbinfo.ro/probleme/1681/power

Să se realizeze un algoritm care să să calculeze a^b.
Exemplu
Dacă a = 5 si b = 4 atunci se va afisa 625 (5^4 = 5 * 5* 5 * 5 = 625)

Bibliografie

[1] **http://www.pbinfo.ro** Descrierea site-ului: "*www.pbinfo.ro îți propune să rezolvi probleme de informatică, cu evaluator automat. Știi pe loc dacă soluția ta este corectă sau dacă trebuie să mai lucrezi la ea.*

Problemele sunt grupate după programa de informatică pentru liceu. Dar nu trebuie să fii la liceu ca să rezolvi aceste probleme. Poți fi elev de gimnaziu, student, profesor sau pur și simplu pasionat de informatică. De fapt, trebuie doar să vrei!!"

[2] **https://www.runceanu.ro/adrian**

[3] Adrian Runceanu, „*Programarea şi utilizarea calculatoarelor*", Editura Academica Brâncuşi din Târgu-Jiu, 2003, ISBN 973-8436-44-3

[4] Adrian Runceanu, Mihaela Runceanu, „*Noțiuni de programare – limbajul C++*", Editura Academica Brâncuși din Târgu-Jiu, 2012, ISBN 978-973-144-550-2

[5] Adrian Runceanu, Mihaela Runceanu, „*Algoritmi implementati in limbajul C++. Volumul I - Algoritmi elementari*", Editura Academica Brâncuși din Târgu Jiu, 2021, ISBN 978-606-9614-06-8