

Algoritmi Fundamentali

Lector dr.
Dorin IORDACHE

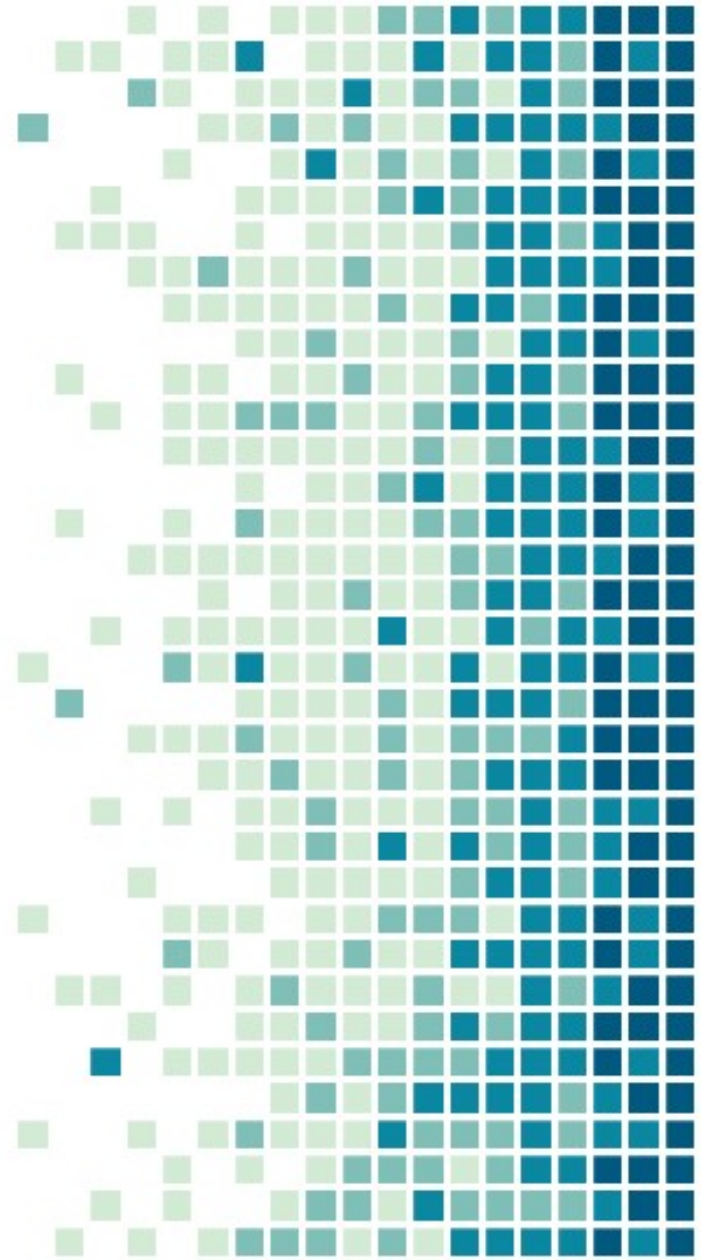


Cursul nr. 5

Algoritmi elementari

Agenda

- Interschimbare
- Maxim / Minim
- Cifrele unui număr
- Divizorii unui număr
- Numere prime
- Cel mai mare divizor comun
- Siruri speciale



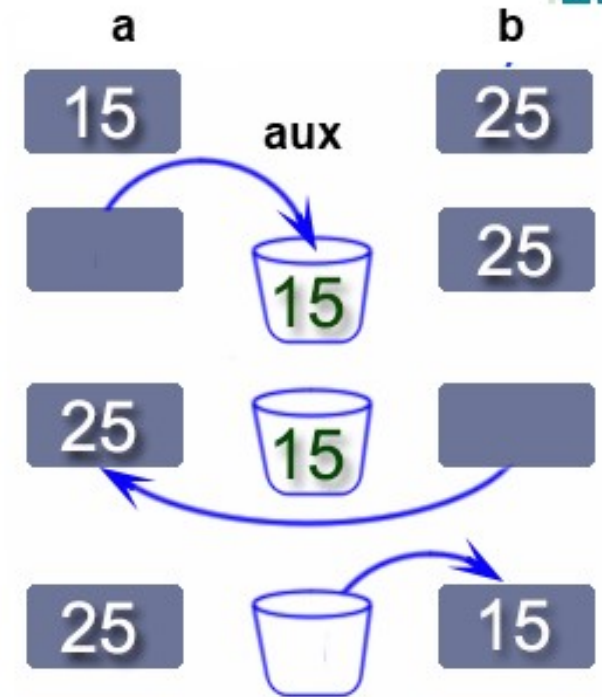


1. INTERSCHIMBARE

1. Interschimbare

Interschimbă (schimbă între ele) valorile a doua variabile de memorie

- Exemplu: Presupunem ca avem 2 pahare:
 - primul pahar (a) conține apă
 - al doilea pahar (b) conține cafea.
 - Cum putem interschimba conținuturile celor două pahare?
Răspuns: utilizand un alt pahar (aux - auxiliar).
- Se vor utiliza 3 variabile: **a** și **b** (variabilele pentru care le vom interschimba valorile) și o variabilă **aux**.



1. Interschimbare

Dacă interschimbăm două valori numerice, putem realiza interschimbarea fără variabilă auxiliară, doar utilizând operații matematice:

- $a \leftarrow a + b;$
- $b \leftarrow a - b;$
- $a \leftarrow a - b;$

Exemplu: $a = 2, b = 3$

- $a = a + b = 2 + 3 = 5;$
- $b = a - b = 5 - 3 = 2;$
- $a = a - b = 5 - 2 = 3;$



Interschimbare

```
integer a,b,aux;  
begin  
    read a,b;  
    write a,b;  
    aux<-a;  
    a<-b;  
    b<-aux;  
    write a,b;  
end.
```

```
int a,b,aux;  
    scanf("%d %d", &a,&b);  
    printf("a=%d , b=%d",a,b);  
    aux=a;  
    a=b;  
    b=aux;  
    printf("a=%d , b=%d",a,b);  
end.
```



A cluster of five small, light blue stars is positioned in the upper left area of the slide, above the main text.

2.DETERMINARE MAXIM / MINIM



Determinare maxim

Să se determine cea mai mare valoare dintr-un șir de n numere întregi citite de la tastatura.

Algoritmul:

- – se vor utiliza variabilele: n (numărul de valori citite), i (contor), a (valoarea citită), \max (maximul)
- – se citește primul număr și se consideră ca fiind cel mai mare ($\max \leftarrow a$)
- – se citesc pe rând valorile de la tastatura și se compară cu maximul curent. Dacă valoarea citită este mai mare decât maximul atunci se atribuie variabilei \max acea valoare.



Determinare maxim

```
integer n,i,a,max;  
begin  
  read n;  
  read a;  
  max <- a;  
  for i <- 2,n do  
    read a;  
    if a>max then  
      max <- a;  
    endif;  
  endfor;  
  write max;  
end.
```

```
int n,i,a,max;  
scanf("%d", &n);  
scanf("%d", &a);  
max = a;  
for(i=1;i<n;i++){  
  scanf("%d", &a);  
  if(max < a)  
    max = a;  
}  
printf("max=%d",max);
```



Determinare maxim

12	45	64	10	13	56
----	----	----	----	----	----

a =	45
-----	----

max < a

a =	64
-----	----

max < a

a =	10
-----	----

max < a

a =	13
-----	----

max < a

a =	56
-----	----

max < a

max=	
------	--

max=	12
------	----

max=	45
------	----

max=	64
------	----

max=	64
------	----

max=	64
------	----

max=	64
------	----

max=	64
------	----



3. PRELUCRAREA CIFRELOR UNUI NUMĂR



a. Extragerea cifrelor numărului

Se citește de la tastatură un număr întreg **nr**. Să se determine suma cifrelor numărului **nr**.

Algoritmul presupune:

- extragerea pe rând a cifrelor numărului **nr** începând de la ultima cifră și
- adăugarea cifrei la sumă prin operația **suma <- suma + nr mod 10** (unde **nr mod 10** este ultima cifră a numărului, apoi
- eliminarea ultimei cifre prin operația **nr <- nr div 10**.

Algoritmul se încheie atunci când s-au adunat toate cifrele numărului, adică valoarea variabilei **nr** este 0.



Suma cifrelor unui numar integer

```
integer nr, suma;  
begin  
  read nr;  
  suma <- 0;  
  while nr <> 0 do  
    suma <- suma + nr mod 10;  
    nr <- nr div 10;  
  endwhile;  
  write suma;  
end.
```

```
int nr, suma;  
scanf("%d", &nr);  
suma = 0;  
while(nr != 0){  
  suma = suma + nr % 10;  
  nr = nr / 10;  
}  
printf("suma=%d", suma);
```



Suma cifrelor unui numar integer

nr=123456

suma = suma + nr % 10

1	2	3	4	5	6
---	---	---	---	---	---

suma =	0
--------	---

nr %10	6
--------	---

nr / 10	12345
---------	-------

nr != 0

suma=	6
-------	---

nr %10	5
--------	---

nr / 10	1234
---------	------

nr != 0

suma=	11
-------	----

nr %10	4
--------	---

nr / 10	123
---------	-----

nr != 0

suma=	15
-------	----

nr %10	3
--------	---

nr / 10	12
---------	----

nr != 0

suma=	18
-------	----

nr %10	2
--------	---

nr / 10	1
---------	---

nr != 0

suma=	20
-------	----

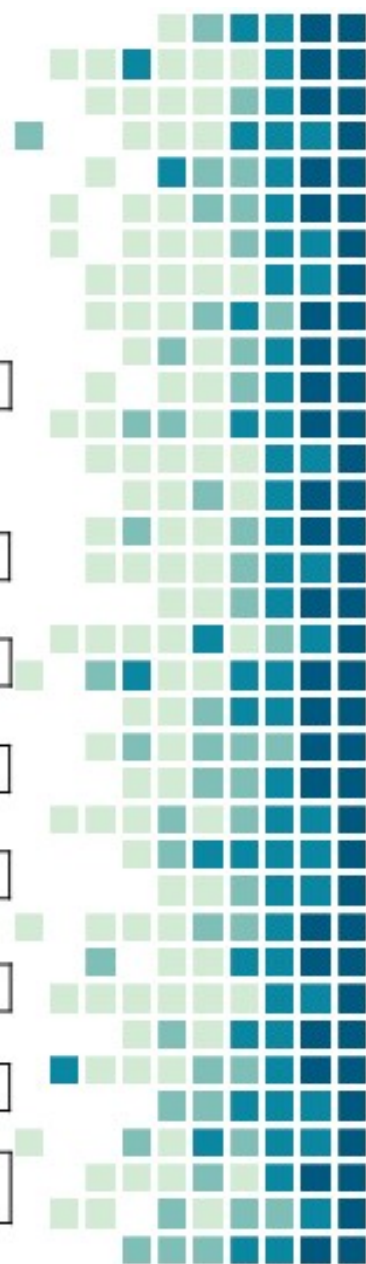
nr %10	1
--------	---

nr / 10	0
---------	---

nr != 0

suma=	21
-------	----

suma =	21
--------	----



b. Inversul unui număr întreg

Se citește de la tastatură un număr întreg **nr**. Să se determine inversul numărului **nr**.

Algoritmul presupune:

- extragerea pe rând a cifrelor numărului **nr** începând de la ultima cifră și
- adăugarea cifrei la suma prin operația **inv <- inv * 10 + nr mod 10** (unde **nr mod 10** este ultima cifră a numărului, apoi
- eliminarea ultimei cifre prin operația **nr <- nr div 10**.

Algoritmul se încheie atunci când s-au adunat toate cifrele numărului, adică valoarea variabilei **nr** este 0.

Iar în variabila **inv** vom obține inversul acestuia.



Inversul unui număr întreg

Observație:

Dacă inversul numărului este egal cu numărul însuși acesta este un **palindrom**.

dacă $inv = nr$ atunci

 write nr este palindrom

end daca



Inversul numărului întreg

```
integer nr, inv;  
begin  
  read nr;  
  inv <- 0;  
  while nr <> 0 do  
    inv <- inv * 10 + nr mod 10;  
    nr <- nr div 10;  
  endwhile;  
  write inv;  
end.
```

```
int nr, inv;  
scanf("%d", &nr);  
inv = 0;  
while(nr != 0){  
  inv = inv * 10 + nr % 10;  
  nr = nr / 10;  
}  
printf("inversul=%d", inv);
```



Inversul numărului întreg

nr=123456

1	2	3	4	5	6
---	---	---	---	---	---

nr %10	6
--------	---

nr / 10	12345
---------	-------

nr != 0

inv =	0
-------	---

nr %10	5
--------	---

nr / 10	1234
---------	------

nr != 0

inv =	6
-------	---

inv =	$6 \cdot 10 + 5 = 65$
-------	-----------------------

nr %10	4
--------	---

nr / 10	123
---------	-----

nr != 0

inv =	$65 \cdot 10 + 4 = 654$
-------	-------------------------

nr %10	3
--------	---

nr / 10	12
---------	----

nr != 0

inv =	$654 \cdot 10 + 3$
-------	--------------------

nr %10	2
--------	---

nr / 10	1
---------	---

nr != 0

inv =	$6543 \cdot 10 + 2$
-------	---------------------

nr %10	1
--------	---

nr / 10	0
---------	---

nr != 0

inv =	$65432 \cdot 10 + 1$
-------	----------------------

inv =	654321
-------	--------

4.DIVIZORII UNUI NUMĂR



Divizorii unui număr

Pentru determinarea divizorilor unui număr se vor utiliza variabilele:
nr (numărul citit de la tastatura), **d** (contor).

- Se vor căuta posibili divizori în intervalul $[2, nr/2]$.
- Dacă **$nr \bmod d = 0$** (nr se divide la d) atunci se afișează divizorul d.



Divizorii unui număr

```
integer nr, d;  
begin  
  read nr;  
  for d <- 2, nr/2 do  
    if nr mod d=0 then  
      write d;  
    endif;  
  endfor;  
end.
```

```
int nr, d;  
scanf("%d", &nr);  
for(d=2; d<=nr/2; d++){  
  if(nr % d == 0)  
    printf("%d, ", d);  
}
```



5.DETERMINARE NUMĂR PRIM



Determinare număr prim

Un număr este prim dacă se divide doar la 1 și la el însuși.

Pentru verificarea unui număr prim se vor utiliza variabilele:

- **nr** (numărul citit de la tastatură), **d** (contor) și **ePrim**.
- Vom considera că numărul **nr** este prim, deci variabila **ePrim** se va inițializa cu valoarea 1. (vom încerca să demonstrăm contrariul). Se vor căuta posibili divizori în intervalul $[2, nr/2]$.
- Dacă **$nr \bmod d = 0$** (nr se divide la d) atunci rezultă că numărul, are divizori, deci nu este prim. Atunci schimbăm valoarea variabilei **ePrim=0**.



Determinare număr prim

```
integer nr, e, ePrim;  
begin  
  read nr;  
  ePrim <- 1;  
  for d <- 2, nr/2 do  
    if nr mod 2=0 then  
      ePrim <- 0;  
    endif;  
  endfor;  
  if ePrim = 1 then  
    write "Numarul este prim";  
  else  
    write "Numarul nu este prim."  
  end.
```

```
int nr, d, ePrim;  
scanf("%d", &nr);  
ePrim=1;  
for(d=2;d<=nr/2;d++){  
  if(nr % d ==0)  
    ePrim=0;  
}  
if(ePrim==1)  
  printf("Numarul este prim);  
else  
  printf("Numarul nu este prim);
```



Determinare număr prim

Ciurul lui Eratosthenes

```
begin Eratosthenes(n)
integer A[n]
for i <- 2,  $\sqrt{n}$  do
  if A[i] is True then
    j <- i2
    while j ≤ n do
      A[j] = False
      j <- j + i
    end while
  endif;
endfor;
return I pentru A[i] True
end Eratosthenes
```

Variante

- **Sundaram**
- **Atkin**

<https://www.baeldung.com/cs/prime-number-algorithms>



Conjectura lui Goldbach

Orice număr natural par, mai mare ca 2, este suma a două numere prime.

Exemple: $4 = 2+2$, $6 = 3+3$, $8=3+5$, $10= 5+5$, etc

```
// Verificare daca un numar este prim
bool estePrim(int n)
{
    if (n <= 1)
        return 0;
    for (int i = 2; i * i <= n + 1; i++)
        if (n % i == 0)
            return 0;
    return 1;
}
```

```
Conjectura Goldbach este adevarata pentru 4
Conjectura Goldbach este adevarata pentru 6
Conjectura Goldbach este adevarata pentru 8
Conjectura Goldbach este adevarata pentru 10
Conjectura Goldbach este adevarata pentru 12
Conjectura Goldbach este adevarata pentru 14
Conjectura Goldbach este adevarata pentru 16
Conjectura Goldbach este adevarata pentru 18
Conjectura Goldbach este adevarata pentru 20
Conjectura Goldbach este adevarata pentru 22
Conjectura Goldbach este adevarata pentru 24
```

```
// Conjectura Goldbach
void ConjecturaGoldbach(int limita)
{
    // Test Goldbach pentru numere [4,limita]
    for (int n = 4; n <= limita; n += 2) {
        int gasit = 0;
        for (int i = 2; i <= n / 2; i++)
            if (estePrim(i) && estePrim(n - i) && gasit == 0)
                gasit = 1;
        if (gasit == 0)
            cout<< "Conjectura Goldbach este falsa pentru  n = "<< n << endl;
        else
            cout << "Conjectura Goldbach este adevarata pentru  "<< n << endl;
    }
}
```


6.DETERMINARE CEL MAI MARE DIVIZOR COMUN



Determinare cel mai mare divizor comun – Algoritmul lui Euclid

Algoritmul determină cel mai mare divizor comun pentru două numere întregi a și b citite de la tastatura.

Se vor utiliza variabilele:

- a, b (numerele citite de la tastatura),
- rest (restul împărțirii întregi, $a \bmod b$).



Determinare cel mai mare divizor comun

```
integer a, b, rest;  
begin  
  read a, b;  
  while b <> 0 do  
    rest <- a mod b;  
    a <- b;  
    b <- rest;  
  endwhile;  
  write a;  
end.
```

```
int a, b, rest;  
scanf("%d %d", &a, &b);  
while(b!=0){  
  rest=a%b;  
  a=b;  
  b=rest;  
}  
printf("cmmdc=%d",a);
```

[Exemplu](#)



7.SIRUL LUI FIBONACCI



Numerele lui Fibonacci

Sirul lui Fibonacci este un sir infinit de numere, care au la baza o formula simpla: $n_2 = n_1 + n_0$. Pe baza acestei formule se genereaza elementele sirului.

Italianul Leonardo of Pisa (cunoscut in matematica drept Fibonacci) a descoperit un sir de numere destul de interesant: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34,

55, 89, 144, 233, 377, ...

Se vor utiliza variabilele:

- n valoarea maximă a elementului din şirul lui Fibonacci
- n_0, n_1, n_2 (numerele inițiale),
- $n_0=0, n_1=1$ și $n_2 = n_0 + n_1$



Generarea numerelor lui Fibonacci

```
integer n0, n1, n2, n;  
begin  
  n0 <- 0;  
  n1 <- 1;  
  n2 <- n0 + n1;  
  write n0, n1, n2;  
  while n2 < n do  
    n0 <- n1;  
    n1 <- n2;  
    n2 <- n0 + n1;  
    write n2;  
  endwhile;  
end.
```

```
int n0,n1,n2,n;  
n0=0;n1=1;  
n2 = n0 + n1;  
printf("%d, %d, %d, ",n0,n1,n2);  
while(n2 < n){  
  n0=n1;  
  n1=n2;  
  n2 = n0 + n1;  
  printf("%d, ",n2);  
}
```



Conjectura Collatz sau Problema din Siracuza

Se spune că dacă începem cu orice pozitiv, să spunem n , atunci fiecare termen se obține din termenul anterior după cum urmează:

- Dacă termenul anterior este par, următorul termen $n/2$;
- Dacă termenul anterior este impar, următorul termen este $3n+1$;
- Se repeta procesul pana cand se obtine 1

```
int Collatz(int n) {  
    int pas = 0;  
    while (n != 1) {  
        if (n % 2 == 0 && n > 1)  
            n /= 2;  
        else  
            n = (n * 3) + 1;  
        pas++;  
    }  
    return pas++;  
}
```

Introdu numarul pentru determinare Collatz?

37

112 56 28 14 7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

Numarul de pasi pentru 37 este 21

Introdu numarul pentru determinare Collatz?

47

142 71 214 107 322 161 484 242 121 364 182 91 274 137 412 206 103 310
155 466 233 700 350 175 526 263 790 395 1186 593 1780 890 445 1336 668
334 167 502 251 754 377 1132 566 283 850 425 1276 638 319 958 479 143
8 719 2158 1079 3238 1619 4858 2429 7288 3644 1822 911 2734 1367 4102
2051 6154 3077 9232 4616 2308 1154 577 1732 866 433 1300 650 325 976 4
88 244 122 61 184 92 46 23 70 35 106 53 160 80 40 20 10 5 16 8 4 2 1

Numarul de pasi pentru 47 este 104



Mulumesc

pentru atenție!

dorin.iordache@365.univ-ovidius.ro