

## Curs h-continuare

### Algoritmul lui Euclid

Calcul Comandc

$$a = 252 = 2^2 \cdot 3^2 \cdot 7$$

$$b = 105 = 3 \cdot 5 \cdot 7$$

$\Rightarrow 3 \cdot 7 = 21 = \text{Comandc factori comuni la puterea maximă}$

$$\begin{array}{r} 252 : 2 \\ 126 : 2 \\ 63 : 3 = 21 \\ 7 : 7 \end{array}$$

$$\begin{array}{r} 105 : 3 \\ 35 : 5 \\ 7 : 7 \\ 1 \end{array}$$

Teorema împărțirii cu rest a numerelor întregi:

$$a = b \cdot q_0 + r_0 \quad \text{deîmpărțit} = \text{împărțitor} \cdot \text{cat} + \text{rest.}$$

$$252 = 105 \cdot ? + ? = 105 \cdot 2 + 42$$

$$\begin{array}{r} 252 : 105 = 2 \\ 210 \\ \hline 42 \end{array}$$

$$\begin{array}{r} 105 : 42 = 2 \\ 84 \\ \hline 21 \end{array}$$

$$105 = 42 \cdot ? + ? = 42 \cdot 2 + 21$$

3 împărțiri

Noul deîmpărțit este împărțitorul curent restul nu vine o.

Noul împărțitor este restul curent

$$42 = 21 \cdot ? + ? = 21 \cdot 2 + 0$$

deîmp. imp.

$$a = b \cdot q_0 + r_0$$

$$b = r_0 \cdot q_1 + r_1$$

$$r_0 = r_1 \cdot q_2 + r_2$$

...

$$r_{m-2} = r_{m-1} \cdot q_m + r_m$$

$$r_{m-1} = r_m \cdot q_{m+1} + 0$$

Comandc(a,b) = rm  
ultimul rest non-zero

$$\begin{array}{r} 210 \\ 42 \\ \hline 252 \end{array}$$

$$b = 105 = 42 \cdot 2 + 21 = (21 \cdot 2) \cdot 2 + 21 = 21 \cdot 4 + 21 = 21 \cdot 5$$

$$a = 252 = 105 \cdot 2 + 42 = (21 \cdot 5) \cdot 2 + 21 \cdot 2 = 21 \cdot 10 + 21 \cdot 2 = 21 \cdot 12$$

Către a,b

int r, cat;

$$r = a \% b;$$

```

while (r != 0)
{
    a = b_i;
    b = r_i;
    r = a % b_i;
}

```

Rezultat: ~~0~~ b

Variantă II. Scăderi repetate

252, 105

$$\begin{array}{r} 252 \\ -105 \\ \hline 147 \end{array}$$

$$\begin{array}{r} 147 \\ -105 \\ \hline 42 \end{array}$$

$$\begin{array}{r} 105 \\ -42 \\ \hline 63 \end{array}$$

$$\begin{array}{r} 63 \\ -21 \\ \hline 42 \end{array}$$

$$\begin{array}{r} 42 \\ -21 \\ \hline 21 \end{array}$$

$$\begin{array}{r} 21 \\ -21 \\ \hline 0 \end{array}$$

6 scăderi

Citire a b<sub>i</sub>

```

while (a != b)
{
    if (a > b) a = a - b;
    else b = b - a;
}

```

Rezultat: a sau b

7. Generarea aleatorie a unor numere

```
#include <time.h>
```

```
int nr = rand(); // random / randomize
```

[0, 32767]

unsigned short

RAND\_MAX

```
int main()
```

```
{ srand(time(NULL)); // start randomize
```

time actual în ms diferentă acum - 01.01.1970  
UNIX

```
int x = rand();
int y = rand();
}
```

3

Maime la accesarea dif.

$$24 \times 60 \times 60 \times 1000 =$$

ms.

Generare valoare în intervalul  $[a, b]$

Citire  $a, b$

$\text{int } x = \text{rand}() \% (b+1);$

$[0, b]$

Se generează o valoare random în intervalul  $[a, b]$  Se se sume cifrele

$$a = 100 \quad b = 1000$$

$$a = 0 \quad b = 900$$

$$[0, 900] \\ [0, 6]$$

$$18537 \% 901 = 517$$

$$a = 100 \quad b = 1000$$

$[a, b]$

$[0, 900]$

$+a$

$\rightarrow [100, 1000]$

$\% 901$

$\text{int } y = a + \text{rand}() \% (b - a + 1);$

8. Măsurarea timpului de execuție

#include <time.h>

int main()

{ clock\_t start, stop;

start = clock();

...

stop = clock();

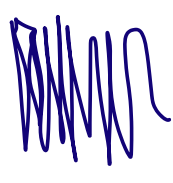
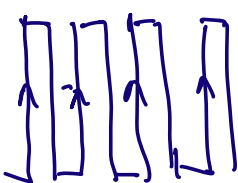
printf("Timpul măsurat %lf", (double)(stop - start) / CLOCKS\_PER\_SEC);

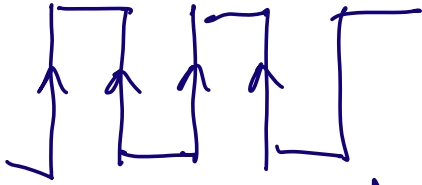
$\rightarrow$  ca și cum unsigned long.

// pornire cronometru

// bucata de cod măsurată fără citiri, afișări

// oprire cronometru





Conversii între baze 10  $\leftrightarrow$  2  $\leftrightarrow$  8  $\leftrightarrow$  16 Net: 67.

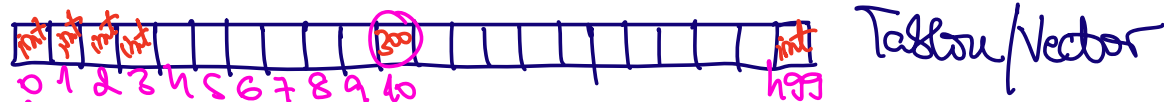
### Curs 5 - Tablouri și alocare relativă

o structură de date care înglobează mai multe variabile de același tip.

Exerciții să se declare 3 variabile ~~500~~ variabile

int a, b, c, d, e, f;

Siruri de numere. Serii de numere



index, indexare, poziție

Formă generală tip nume [capacitate maximă],

int t[500];

char c[255];

În C/C++ tablourile se numărătoare/se indexează

de la 0 și până la cap maximă - 1.

t[10] = 300;

↳ referința prin indice la un anumit element  
cu atributul 300.

Elementul al 0-lea este cel din stânga element

Elementul cap. max - 1 este cel din dreapta element.

Tablou = Vector = Array = 1D array

Citirea elementelor elemente ale tabloului:

int n[10];  
int a;  
citire repetind a tuturor elementelor  
for  
lungimea efectivă a tabloului

printf("Dati lungimea tabloului n<=10");

scanf("%d", &n); // n=5

for (int i=0; i<n; i++)  
{ printf("n[%d]= ", i);  
scanf("%d", &n[i]);  
}

i=0 i=1 i=2  
n[0]=70 n[1]=-35 n[2]=50  
n[3]=0 n[4]=100

Afisarea elementelor tabloului

printf("Elementele tabloului sunt:");  
for (int i=0; i<n; i++)

printf("%d ", n[i]);

printf("\n");

70 -35 50 0 100

Initializarea tabloului

float n[5] = {1.2f, 3.4e3f, -2f, 90.7e-8f, -88.5e-7f};

1.2 3.4 · 10<sup>3</sup> -2 90.7 · 10<sup>-8</sup> -88.5 · 10<sup>-7</sup>  
-88.5 · 10<sup>-7</sup> = -88.5 / 1000000 = 0.0000885

constanta float

int x[5] = {1, 2};

x[0]=1 x[1]=2 x[2]=0 x[3]=0 x[4]=0

1 2

Initializarea elementelor cu 0

`int a[100] = {0};` `a[0] = 0` restul automat 0.  
intentionat

`int a[];` // ~~gresit~~ corect

Numărul de elemente

`int x[] = {1, 2, 3, 4, 5};`

`int n = sizeof(x) / sizeof(x[0]);`  $n=5$ .  
20 Bytes  
4 Bytes  
fiecare int 4B.

`int x[100];`

Matrice (Tablou 2D, Tablou bidimensional)

$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$   $\begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix}$

$\begin{pmatrix} 7 & 3 & 6 \\ 20 & -2 & 0 \end{pmatrix}$

`int a[20][30];`

`int m, n;`

→ nr. efectiv de coloane  
→ nr. efectiv de linii

`printf("Dati nr. linii si nr. coloane n <= 20, m <= 30");`

`scanf("%d%d", &n, &m);`  $n=3, m=4$ .

`for (int i=0; i<m; i++)`

`for (int j=0; j<n; j++)`

`printf("a[%d][%d] = ", i, j);`

`scanf("%d", &a[i][j]);`

Trec prin fiecare linie de m ori  
si prin fiecare coloana de n ori  
toate elementele

$i=0$  `a[0][0] =`  
`a[0][1] =`  
`a[0][2] =`  
`a[0][3] =`

0 1 2 3 ← coloane  
0 1 2 3  
5 6 7 8

0 1 2 3  
0 1 2 3 4  
1 5 6 7 8  
2 9 10 11 12

29

Figure

```
printf("Matricea este:");  
for (int i=0; i<=m; i++)  
{  
    for (int j=0; j<=n; j++)  
        printf("%d\t", a[i][j]);  
    printf("\n");  
}
```

i=1    a[1][0] = —  
         a[1][1] = —  
         a[1][2] = —  
         a[1][3] = —

1 2 3 4 5 6 7 8 9 10 11 12  
1 2 3 4 5  
5 6 7 8 9  
9 10 11 12 13