

Εισαγωγή στην Πληροφορική & στον Προγραμματισμό

Αρχές Προγραμματισμού Η/Υ (με τη γλώσσα C)

Α' Ανακεφαλαίωση

Παναγιώτης Παύλου

Πέμπτη, 14 Απριλίου 2022

c-programming-22@allos.gr

Ανακεφαλαίωση

Μαθημάτων 1-5

```
int f(int x) {
```

```
    if (x < 0) {  
        return false; -1;  
    }  
    [else  
     return ... ]
```

}

~~~~~  
~~~~~  
~~~~~  
~~~~~

```
bool g(x) {
```

~~~~~  
~~~~~  
~~~~~

```
    if (....)  
        return true;  
    else  
        return false;
```

Ⓜ

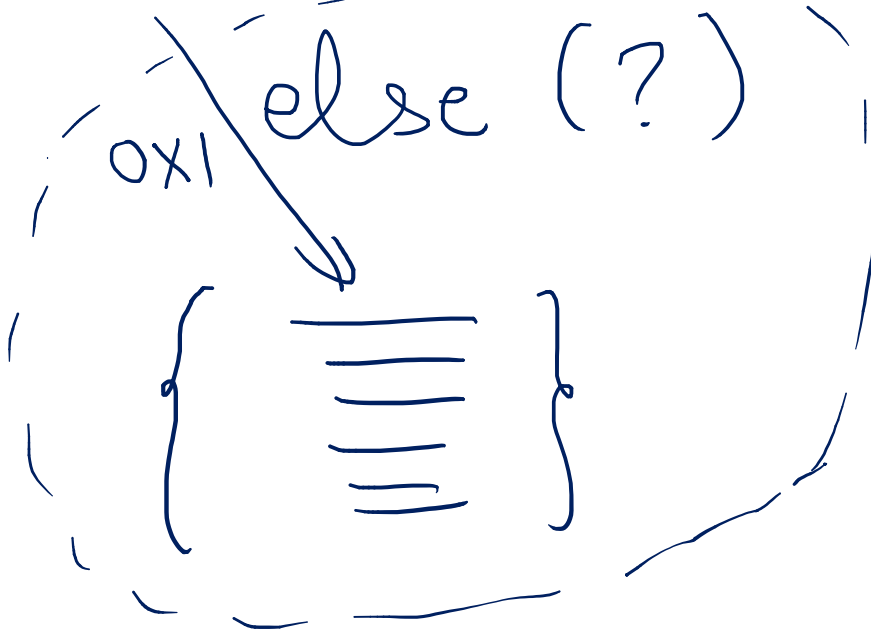
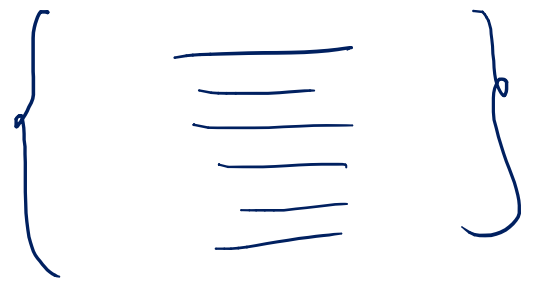
NAI / OXI

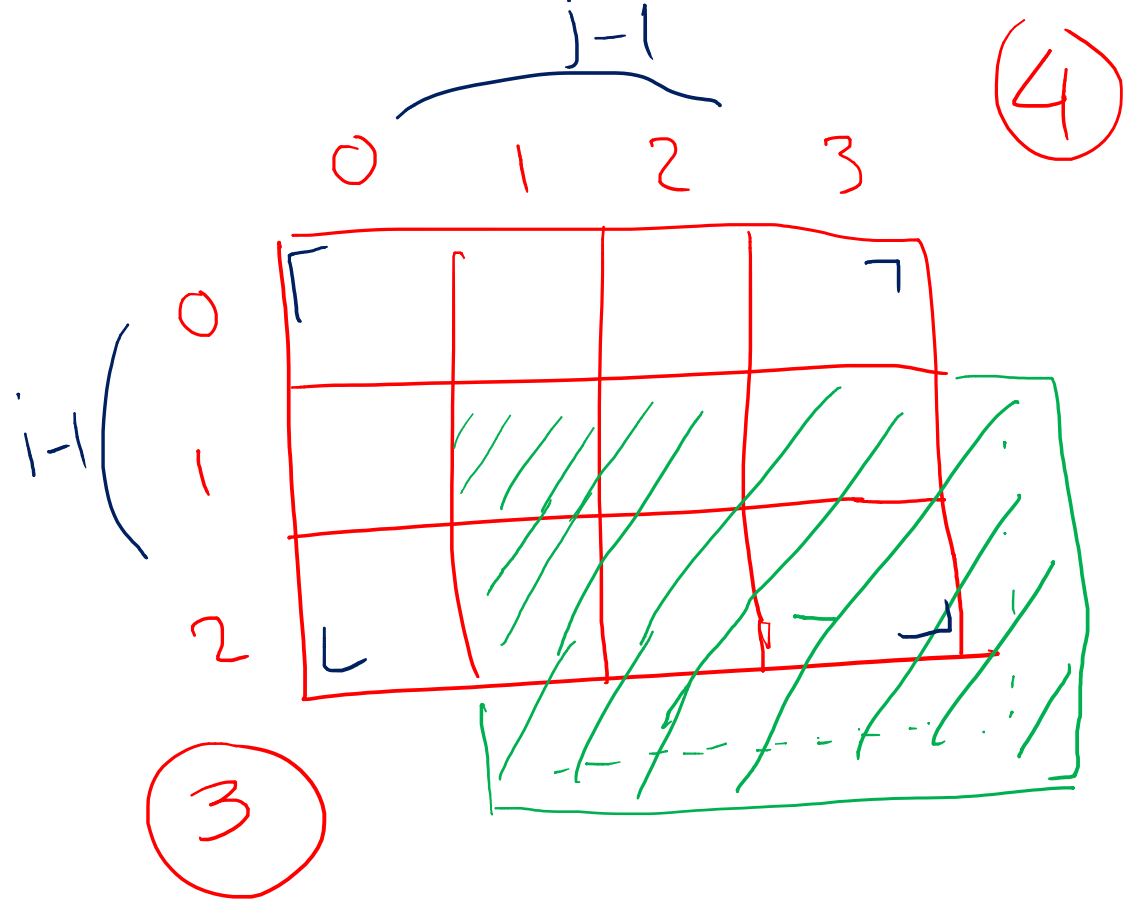
if ( \_\_\_\_\_ )

NAI

OXI

else (?)





```
for (  $i=1$ ;  $i \leq 3$ ;  $i++$ )
for (  $j=1$ ;  $j \leq 4$ ;  $j++$ )
```

$array[i][j]$

$array[\underbrace{i-1}_{0-2}][\underbrace{j-1}_{0-3}]$





$x$   
 $y$   
 $a[2]$   
 $b[i]$

$1+2$   
 $x+5$   
 $y$   
 $y/12$

$1+2$   
 $x+5$

$y$   
 $y/12$

~~$1+1$~~   
 ~~$x+1$~~

$while(x = y/2) \text{ if } (x)$

$x+10$   
 $f(y)$

$false \equiv 0$   
 $true \neq 0$

$-1$   
 $+1000$   
 $\pi$

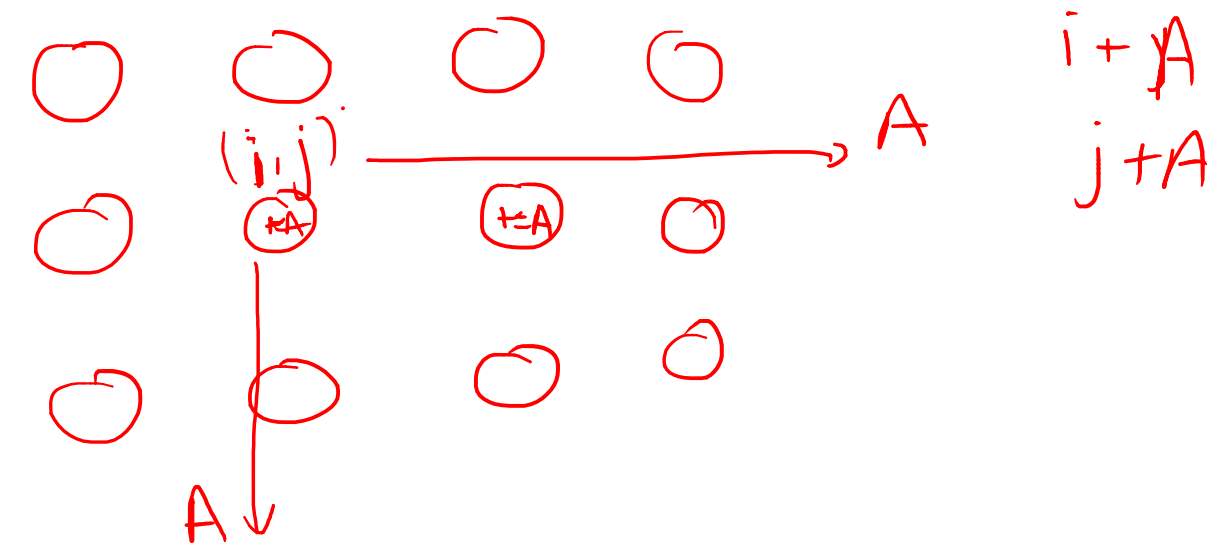
$while((x = y/2) > 3)$

$if(x != 0)$

$$\overline{w} = \overline{w} + w' \quad \Bigg| \quad \overline{w} + w' = \overline{w}$$

A

⇓



switch ( A ) {

A X Y Z

→ case X :

- case Y :

→ default :

- case Z :

①

↓ break return



# Εφαρμογές #1

Αλγεβρικοί πίνακες

# Δημιουργία συναρτήσεων αλγεβρικών πινάκων

Γράψτε τις παρακάτω συναρτήσεις:

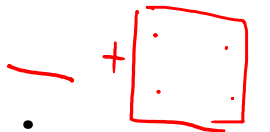
```
void printArray(int M, int N, int array[M][N]);
```

```
void increaseBy(int A, int M, int N, int array[M][N]);
```

```
long sumOfArray(int M, int N, int array[M][N]);
```

$a_{11}, a_{12}$

$a_{21}, a_{22}$



# Εφαρμογές #2

Μονοδιάστατοι πίνακες – [Filter](#) , [Map](#) , [Reduce](#)

ή

Πώς να αντιμετωπίζετε παρόμοια ζητούμενα που αφορούν μονοδιάστατους πίνακες

$\sum x_i^2$   $\sum x_i^2 + i \cdot (x_i > 0)$  int  $\times$  [100];

Δημιουργία συναρτήσεων μονοδιάστατων πινάκων  $i$

Java JS Python ...

Γράψτε τις παρακάτω συναρτήσεις:

`int filterPositive(int L, int arrInput[L], int arrOutput[L]);`

`int mapToSquares(int L, int arrInput[L], int arrOutput[L]);`

$0 - (L-1)$   $L$   $L+1$

`int reduceToSum(int L, int arrInput[L], int arrOutput[L]);`

void

