

Εισαγωγή στην Πληροφορική & στον Προγραμματισμό

Αρχές Προγραμματισμού Η/Υ (με τη γλώσσα C)

Διάλεξη #5

Παναγιώτης Παύλου

Παρασκευή, 2 Απριλίου  2021

c-programming@allos.gr

Ανακεφαλαίωση

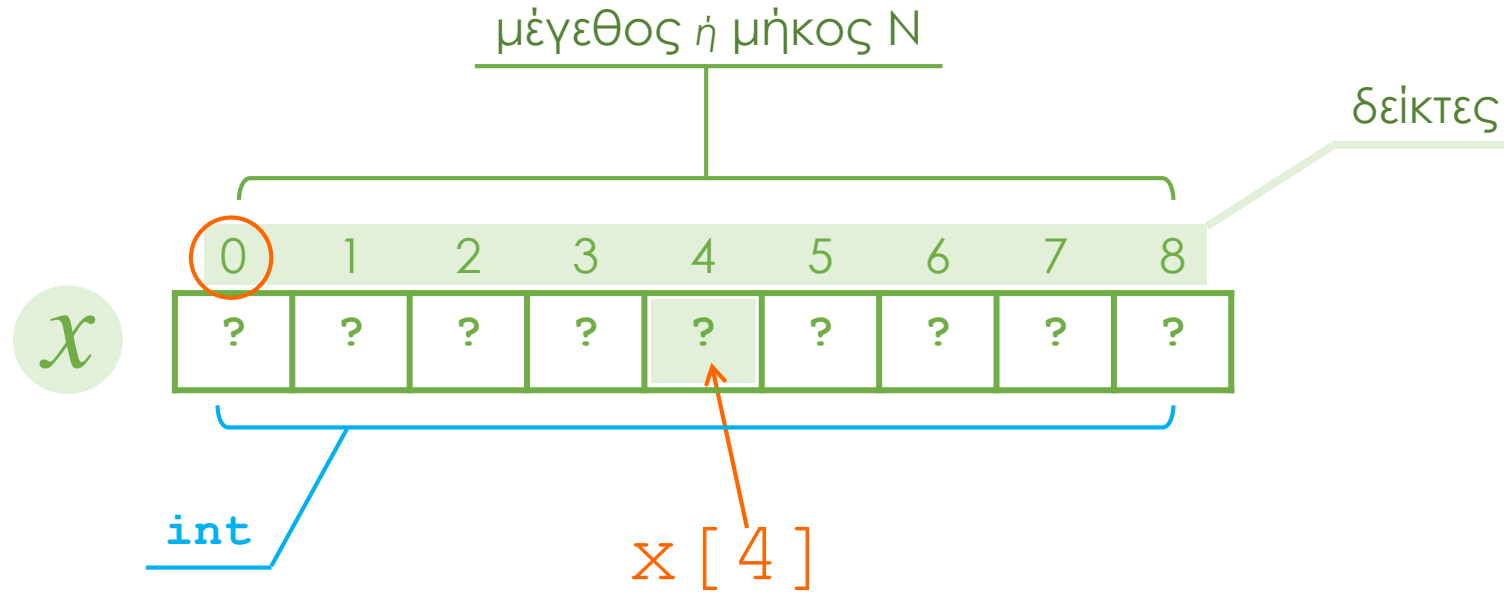
Με μία ματιά να θυμηθούμε τη θεωρία

break και continue

```
for (int i=0; i<N; i++) {  
    ...  
    if (...) {  
        continue;  
    }  
    ...  
    if (...) {  
        break;  
    }  
    ...  
}
```

Πίνακες

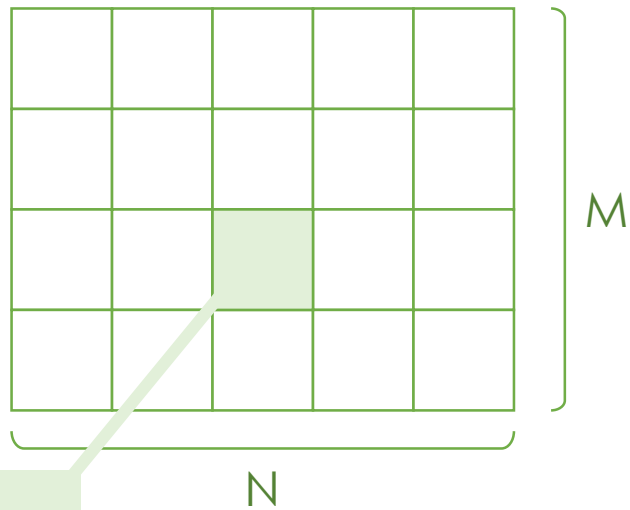
A 8



Απλή δήλωση int x[N] = { 1, 3, 5, 7, 8 };

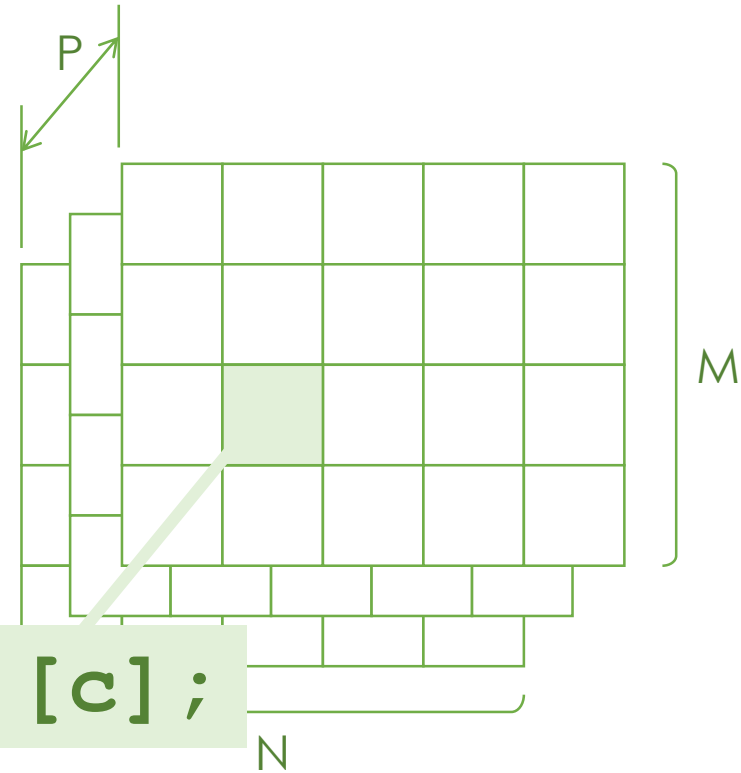
Δήλωση με αρχικοποίηση

Πίνακες 2+ διαστάσεων



```
x[r][c];
```

```
int x[M][N];
```



```
x[p][r][c];
```

```
int x[P][M][N];
```

Εφαρμογές

Ας εφαρμόσουμε επιτέλους τη θεωρία

Πρακτική 1 – Πρώτοι αριθμοί



Υπολογισμός πρώτων αριθμών με το Κόσκινο του Ερατοσθένη

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

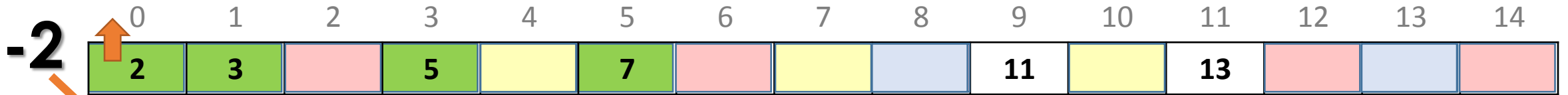
Η μεθοδολογία αυτή θα δούμε εδώ πως εφαρμόζεται για τον υπολογισμό των πρώτων αριθμών, όμως έχει αρκετά ευρείες εφαρμογές. Η λογική είναι από το σύνολο των πιθανών τιμών, με απλό τρόπο να απορρίψουμε όλες αυτές που δεν αποτελούν λύσεις (εδώ όσους δεν είναι πρώτοι). Έτσι όσες τιμές περισσέψουν είναι οι ζητούμενες λύσεις.

Αυτό στους πρώτους αριθμούς από το 1 μέχρι το 100 μεταφράζεται σε απόρριψη όλων των πολλαπλασίων του 2, του 3 και γενικά του επόμενου μη διαγραμμένου κοκ μέχρι εκεί που έχει νόημα ($=\sqrt{100}$).

Προσοχή! Δεν απορρίπτουμε τα 1×2 ή 1×3 αλλά μόνο τα πολλαπλάσιά τους.

Ας γράψουμε τη συνάρτηση `void printPrimesTo (N)` εκτυπώνει τους πρώτους αριθμούς ως το N

Πρώτοι αριθμοί - παρατηρήσεις



$i : 0 \ 1 \ 2 \ 3 \ \dots$

$i+1 : 1 \ 2 \ 3 \ 4 \ \dots$

$2 : 2*(i+1) - 2 \rightarrow (4 \ 6 \ 8 \ 10 \ 12 \ \dots) - 2 \rightarrow 2 \ 4 \ 6 \ 8 \ 10 \ \dots$

$3 : 3*(i+1) - 2 \rightarrow (6 \ 9 \ 12 \ 15 \ \dots) - 2 \rightarrow 4 \ 7 \ 10 \ 13 \ \dots$

$5 : 5*(i+1) - 2 \rightarrow (5 \ 10 \ 15 \ \dots) - 2 \rightarrow 3 \ 8 \ 13 \ \dots$

Πρακτική 2 – Τρίλιζα



Εάν σε έναν πίνακα 3x3 θέσουμε τα στοιχεία του ως κενά ή να περιέχουν τα σύμβολα X ή O τότε μπορούμε να παραστήσουμε μία τρίλιζα!

Στην τρίλιζα ξεκινά πάντα να παίζει πρώτος ο X (το θεωρούμε δεδομένο). Ας δούμε μερικές συναρτήσεις που έχουν ενδιαφέρον:

1. `int playersTurn(int board[3][3])`

Θα επιστρέφει τον παίκτη που έχει σειρά να παίξει ή -1 ως ένδειξη λάθους.

2. `bool tripletExists(int board[3][3], int player)`

Θα επιστρέφει αληθές εάν ο παίκτης player έχει κάνει τρίλιζα.

3. `bool canPlay(int board[3][3], int row, int col, int player)`

Θα επιστρέφει αληθές αν ο παίκτης player επιτρέπεται να παίξει στη θέση row,col

4. `bool playAt(int board[3][3], int row, int col)`

παίξει τον παίκτη που είναι η σειρά του, στη θέση row,col , αλλιώς επιστρέφει false

Τρίλιζα παρατηρήσεις

	0	1	2
0	X	O	O
1	O	X	
2	X	X	O

	0	1	2
0			
1			
2			

	0	1	2
0			
1			
2			

x012
y012

	0	1	2
0			
1			
2			

x012
y210

	0	1	2
0			
1			
2			

x000
y012

	0	1	2
0			
1			
2			

x012
y000

Ερωτήσεις?

- Διαβάστε τις σημειώσεις, διαβάστε τις διαφάνειες και δείτε τα videos **πριν** ρωτήσετε
- **Συμβουλευτείτε** τη σελίδα ερωταποκρίσεων του μαθήματος

<https://qna.c-programming.allos.gr>

- **Στείλτε** τις ερωτήσεις σας πριν και μετά το μάθημα στο

c-programming@allos.gr

- Εάν έχετε **πρόβλημα** με κάποιο κώδικα στείλτε μαζί τον κώδικα και τα μηνύματα λάθους από το CLI ως κείμενα με copy/paste. Εάν θεωρείτε ότι επιπλέον βοηθά και ένα στιγμιότυπο οθόνης, είναι καλοδεχούμενο.
- Επαναλαμβάνουμε : Μην στείλετε ποτέ κώδικα ως εικόνα μας είναι παντελώς άχρηστος!

