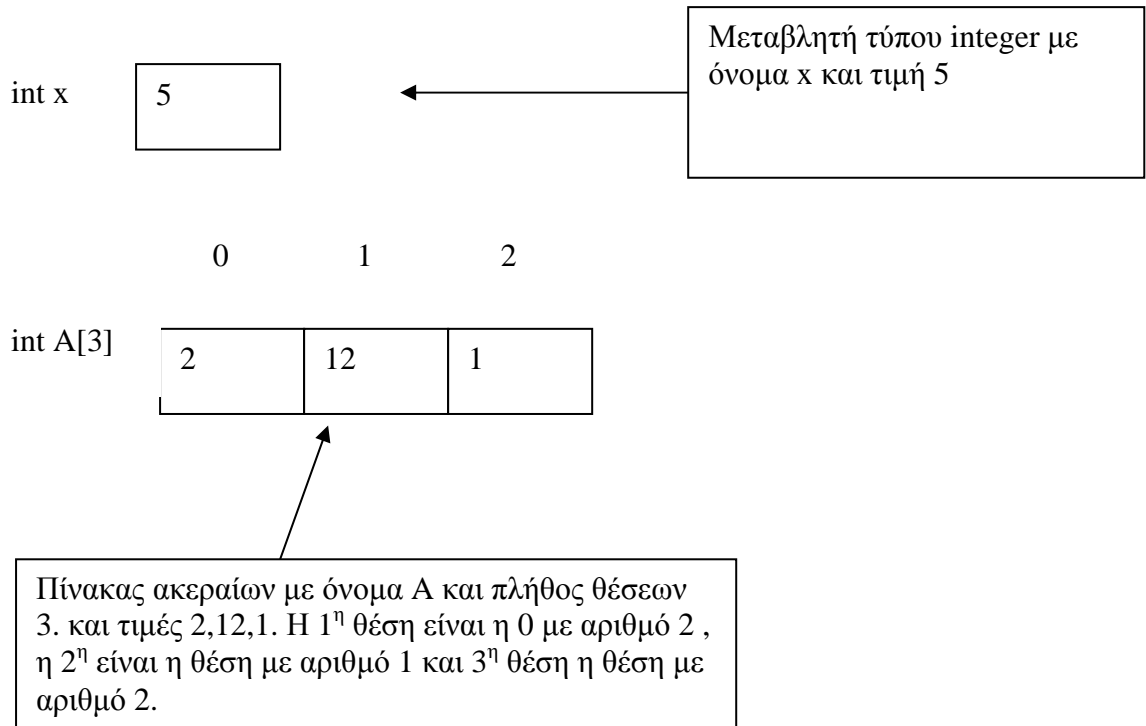


Μάθημα 5^ο Πίνακες

1. Τι είναι πίνακας ;

Πίνακας είναι πολλές μεταβλητές ίδιου τύπου που τις δεσμεύω όλες μαζί. Για να το καταλάβουμε αυτό δίνω τα παρακάτω 3 παραδείγματα:



2. Πώς δηλώνω έναν πίνακα.

int myarray[10];// πίνακας 10 θέσεων 0..9 με περιεχόμενα ακεραίους και όνομα myarray

float array2[5];

πίνακας 5 θέσεων από 0..4 με όνομα array2 και περιεχόμενα πραγματικούς αριθμούς

3. Πώς αρχικοποιώ έναν πίνακα.

Όταν θέλω να αρχικοποιήσω μία απλή μεταβλητή γράφω x=4. Με την προϋπόθεση ότι την έχω δηλώσει νωρίτερα. Για έναν πίνακα η δήλωση και η αρχικοποίηση γίνονται για πρακτικό λόγο μαζί αν και μπορούν να γίνουν ένα-ένα ξεχωριστά:

int testArray[5]={3,5,2,8,7};//πίνακας ακεραίων 5 θέσεων 0..4 με όνομα testArray

char myname[7]='m','i','h','a','l','i','s';//πίνακας χαρακτήρων 8 θέσεων (0..7) με όνομα myname

.....

4. Πώς γεμίζω έναν πίνακα με στοιχεία από το πληκτρολόγιο

Πχ Γέμισμα πίνακα ακεραίων :

```
int myarray[5];
int thesi;
for(thesi=0;thesi<=4;thesi++)
{
    printf("Pliktrologise to %d stoixeio", thesi);
    scanf("%d",&A[thesi]);
}
```

5. Πώς εμφανίζω τον πίνακα ακεραίων μου μόλις γέμισα.

```
for(thesi=0;thesi<=4;thesi++)
{
    printf("%d\n", A[thesi]);
}
```

Αν μας ζηταγε να το εμφανίσουμε αντίστροφα τι θα κάνουμε;

6. Πώς θα εμφανίσω το στοιχείο της θέσης με αριθμό 2 (δηλαδή της 3^{ης} θέσης)

```
printf("%d",A[2]);
```

7. Πώς θα εκχωρίσω τιμή στο 2^ο στοιχείο του πίνακα (δηλαδή αυτό με αριθμό θέσης 1) την τιμή =31 και να εμφανίσω την τιμή του νέου στοιχείου.

```
A[1]=31;
printf("%d",A[1]);
```

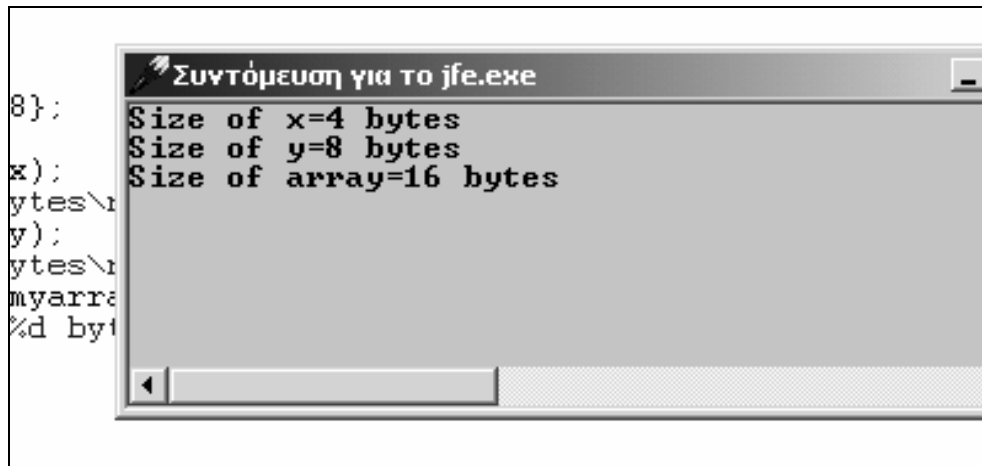
8. Τι κάνει η συνάρτηση sizeof(a) (οπου a είναι το όνομα μιας μεταβλητής ή ενός πίνακα)

Η συνάρτηση sizeof(a) μας επιστρέφει το μέγεθος σε bytes της μεταβλητής ή ενός πίνακα που παίρνει σαν όρισμα. πχ

```
int x=34;
int megethos_bytes;
megethos_bytes=sizeof(x);
printf("%d", megethos_bytes) ; //εμφανίζει 4 .Αρα 4 bytes μέγεθος μεταβλητής
```

Τι θα εμφανίσει το παρακάτω πρόγραμμα;

```
#include<stdio.h>
void main()
{
int x=34;
double y=33;
int myarray[4]={2,6,4,8};
int megethos_bytes;
megethos_bytes=sizeof(x);
printf("Size of x=%d bytes\n", megethos_bytes);
megethos_bytes=sizeof(y);
printf("Size of y=%d bytes\n", megethos_bytes);
megethos_bytes=sizeof(myarray);
printf("Size of array=%d bytes\n", megethos_bytes);
}
```

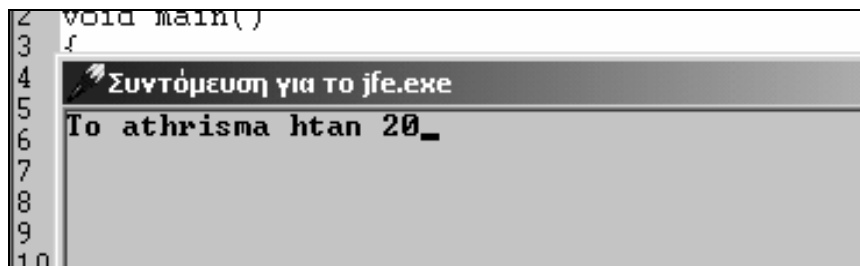


Αρα τι θα μας εμφανίσει η εντολή sizeof(myarray)/sizeof(myarray[0]) και γιατί;

9. Να γίνει πρόγραμμα που να γεμίζει έναν πίνακα 5 θέσεων με τους αριθμούς 4,2,6,7,1 και να εμφανίσει το άθροισμά των τιμών των στοιχείων του πίνακα.

```
#include<stdio.h>
void main()
{
int myarray[5]={4,2,6,7,1};
int i,sum=0;
for(i=0;i<=4;i++)
{
sum=sum+myarray[i];
}

printf("To athrisma htan %d",sum);
}
```

A screenshot of a C program execution. The code is shown in a text editor with line numbers 2 to 10. The program defines an array myarray with values {4, 2, 6, 7, 1} and calculates their sum. The output window shows the message "Συντόμευση για το jfe.exe" and the result "To athrisma htan 20_".

```
2 void main()
3 {
4     Συντόμευση για το jfe.exe
5     To athrisma htan 20_
6
7
8
9
10
```

Αν μας ζήταγε το γινόμενο των στοιχείων του πίνακα τι θα κάναμε;

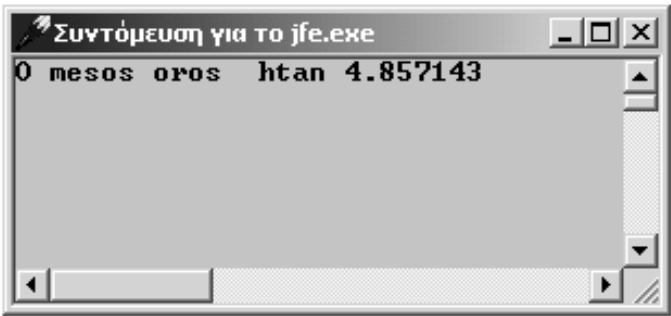
10. Να γίνει πρόγραμμα που να γεμίζει έναν πίνακα 5 θέσεων με αριθμούς από το πληκτρολόγιο και να εμφανίσει το άθροισμά των τιμών των στοιχείων του πίνακα.

```
#include<stdio.h>
void main()
{
int myarray[5]={4,2,6,7,1};
int i,sum=0;
for(i=0;i<=4;i++)
{
printf("Pliktrologise to %do stoixeio\n",i+1);
scanf("%d",&myarray[i]);
sum=sum+myarray[i];
}
printf("To athrisma htan %d",sum);
}
```

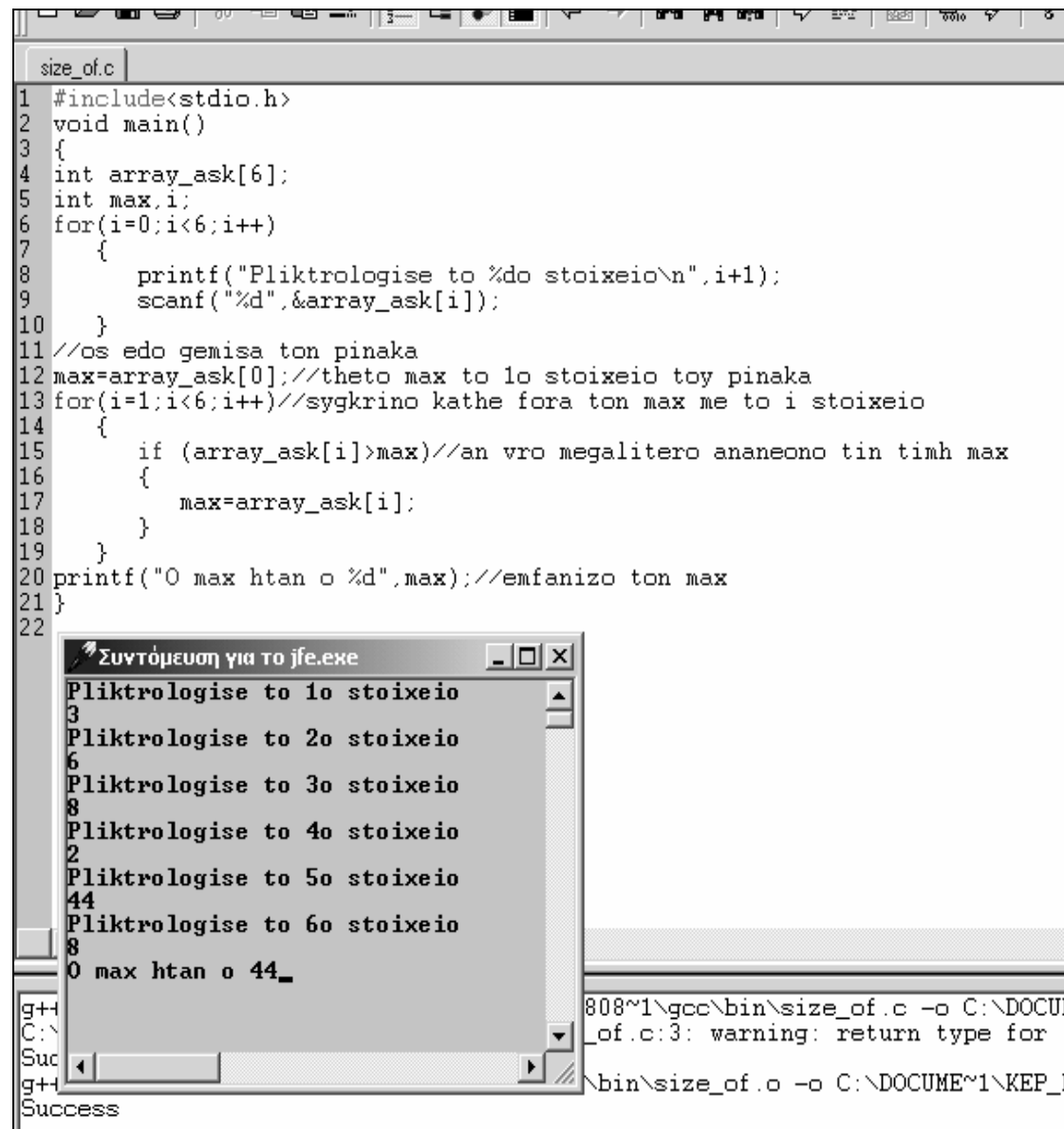
```
size_of.c
1
2
3 Πλκτροlogise to 1o stoixeio
4 2
5 Πλκτροlogise to 2o stoixeio
6 3
7 Πλκτροlogise to 3o stoixeio
8 5
9 Πλκτροlogise to 4o stoixeio
10 4
11 Πλκτροlogise to 5o stoixeio
12 6
13 To athrisma htan 20_
14
15
16
17
```

**11. Να γίνει πρόγραμμα που να βρίσκει το μέσό όρο του πίνακα int
arrayB[7]={2,4,6,8,3,5,6}**

```
size_of.c
1 #include<stdio.h>
2 void main()
3 {
4 int arrayB[7]={2,4,6,8,3,5,6};
5 int i,sum=0;
6 for(i=0;i<=6;i++)
7 {
8     sum=sum+arrayB[i];
9 }
10 printf("0 mesos oros htan %lf",sum/7.0);
11 }
12
```



Να γίνει πρόγραμμα που να βρίσκει τη μέγιστη τιμή σε έναν πίνακα `int array_ask[6]` με στοιχεία που θα έχει δώσει ο χρήστης από το πληκτρολόγιο.



```
size_of.c
1 #include<stdio.h>
2 void main()
3 {
4     int array_ask[6];
5     int max,i;
6     for(i=0;i<6;i++)
7     {
8         printf("Pliktrologise to %do stoixeio\n",i+1);
9         scanf("%d",&array_ask[i]);
10    }
11    //os edo gemisa ton pinaka
12    max=array_ask[0]; //theto max to 1o stoixeio toy pinaka
13    for(i=1;i<6;i++) //sygkrino kathe fora ton max me to i stoixeio
14    {
15        if (array_ask[i]>max) //an vro megalitero ananeono tin timh max
16        {
17            max=array_ask[i];
18        }
19    }
20    printf("O max htan o %d",max); //emfanizo ton max
21 }
22
```

```
Συντόμευση για το jfe.exe
Pliktrologise to 1o stoixeio
3
Pliktrologise to 2o stoixeio
6
Pliktrologise to 3o stoixeio
8
Pliktrologise to 4o stoixeio
2
Pliktrologise to 5o stoixeio
44
Pliktrologise to 6o stoixeio
8
O max htan o 44_
```

```
g++ C:\808~1\gcc\bin\size_of.c -o C:\DOCUM~1\KEP_
of.c:3: warning: return type for
bin\size_of.o -o C:\DOCUME~1\KEP_
Success
```

Αν μας ζήτηγε τον ελάχιστο του πίνακα τι θα κάναμε;

Παράδειγμα με γέμισμα πίνακα χαρακτήρων 4 θέσεων. Προσοχή ο πίνακας δεν είναι αλφαριθμητικό (string) όπως θα δούμε στα επόμενα μαθήματα. Προσέχω τη λειτουργία της scanf() γιατί όταν συνοδεύεται από χαρακτήρα μεμονωμένο παίρνει μαζί της και το χαρακτήρα νέας γραμμής στον buffer. Έτσι λοιπόν αν προσπαθήσω να γεμίσω έναν πίνακα χαρακτήρων και να εμφανίσω τα στοιχεία του παίρνω απρόσμενα αποτελέσματα. Η παρακάτω υλοποίηση διορθώνει το πρόβλημα και εμφανίζει τα στοιχεία σωστά.

```
1 #include<stdio.h>
2 void main()
3 {
4     char myarray[4];
5     int thesi;
6     for(thesi=0;thesi<=3;thesi++)
7     {
8         printf("Pliktrologise to %do stoixeio\n",thesi+1);
9         scanf("\n%c", &myarray[thesi]);
10    }
11    printf("Ta stoixeia tou pinaka einai\n");
12    for(thesi=0;thesi<=3;thesi++)
13    {
14        printf("%c",myarray[thesi]);
15    }
16 }
17
```

Οποσδήποτε με αλλαγή γραμμής

jfe - Συντόμευση

Pliktrologise to 1o stoixeio
m
Pliktrologise to 2o stoixeio
i
Pliktrologise to 3o stoixeio
k
Pliktrologise to 4o stoixeio
i
Ta stoixeia tou pinaka einai
miki_

Το παραπάνω πρόβλημα μπορεί να διορθωθεί και με τη χρήση της getchar() η οποία αν τοποθετηθεί μετά την κλήση της scanf() θα μας απαλλάξει από τον κενό χαρακτήρα που έχει μείνει στον buffer πιο πριν και το πρόγραμμα θα συνεχιστεί κανονικά. Το γέμισμα πινάκων από το πληκτρολόγιο είναι ένα παράδειγμα που αποφεύγεται να διδάσκεται και για το λόγο αυτό λίγες φορές θα το συναντήσουμε σε βιβλία κτλ., οφείλουμε όμως να ξέρουμε πως γίνεται.

Αλγόριθμοι Ταξινόμησης –Αναζήτησης σε πίνακες

Σειριακή ή ακολουθιακή αναζήτηση στοιχείου σε πίνακα

Έστω ότι μας δίνεται πίνακας `int A[7]={3,5,1,8,90,11,-3}` και μας ζητείται να βρούμε αν υπάρχει ένας αριθμός πχ αν υπάρχει ο αριθμός 8. Τότε σύμφωνα με τον αλγόριθμο της ακολουθιακής ή σειριακής αναζήτησης θα διασχίσω τον πίνακα όλο από την αρχή και θα ελέγχω αν το ζητούμενο στοιχείο είναι ίσο με το στοιχείο της εκάστοτε θέσης του πίνακα. Λαμβάνω υπόψη 3 περιπτώσεις:

Α)το στοιχείο να μην υπάρχει καθόλου

Β)το στοιχείο να υπάρχει μία φορά

Γ)το στοιχείο να υπάρχει περισσότερες φορές

Παράδειγμα:

```
serial_search.c
1 /*Seiriakh anazhtsh*/
2 #include<stdio.h>
3 void main()
4 {
5     int A[7]={3,5,1,8,90,11,-3};
6     int zitoumenos,thesi,vrethike;
7     printf("Pliktrologhse zitoumeno arithmo\n");
8     scanf("%d",&zitoumenos);
9     vrethike=0;//theoro oti den yparxei to stoixeio moy
10    for(thesi=0;thesi<=6;thesi++)//diasxizo olo ton pinaka
11    {
12        if(zitoumenos==A[thesi])//sygkrino zitoymeno me stoixeio pinaka
13        {
14            printf("Vrethike sthn thesi=%d\n",thesi);
15            vrethike=1;//an yparxei tha ginei 1 h vrethike
16        }
17    }
18    if(vrethike==0)//ean den vrethike
19    {
20        printf("To stoixeio sas den vrethike\n");
21    }
22 }
23
```

Δυναδική ή διχοτομική αναζήτηση στοιχείου σε πίνακα.

Η διχοτομική ή δυαδική αναζήτηση εφαρμόζεται μόνο σε ταξινομημένο πίνακα (φθίνουσα ή αύξουσα σειρά). Δεν εφαρμόζεται ποτέ σε μη ταξινομημένο πίνακα. Έστω ο πίνακας με τους αριθμούς $\text{int } B[9]=\{100,88,75,43,22,12,9,4,1\}$ και αναζητώ το στοιχείο 9. Παρατηρώ ότι η μεσαία θέση του πίνακα είναι η θέση με αριθμό 4 δηλαδή η 5^η θέση $(0+8)/2=4$. Είναι δηλαδή το στοιχείο 22. Το 9 δεν είναι ίσο με το 22. Το ζητούμενο 9 είναι μικρότερο από το στοιχείο της θέσης με αριθμό 4 που σημαίνει ότι θα βρίσκεται δεξιά από το 22. Άρα το αναζητώ στον υποπίνακα που ορίζεται από τα στοιχεία 12,9,4,1. Ξαναβρίσκω το νέο μέσο του υποπίνακα 12 9 4 1 και ξαναελέγχω που βρίσκεται το ζητούμενο 9. Ο αλγόριθμος «σπάει» συνέχεια τον πίνακα στη μέση μέχρι να βρεί το ζητούμενο ότι είναι ίδιο με το μέσο που έχει υπολογίσει εκείνη τη στιγμή. Λαμβάνω και εδώ κάποιες περιπτώσεις υπόψη:

α) το στοιχείο να μην υπάρχει καθόλου γιατί είναι εκτός των ορίων .

πχ στον παραπάνω πίνακα να ψάξω για το 120 ή για το -10

β) το στοιχείο να είναι μέσα στα όρια του πίνακα , αλλά να μην υπάρχει πχ στον παραπάνω πίνακα να έψαχνα για το 23

γ) το ζητούμενο στοιχείο να υπάρχει περισσότερες από μία φορές πχ στον παραπάνω πίνακα να είχα το 88 δύο φορές.

Παράδειγμα:

100 88 75 43 22 12 9 4 1

$\text{arxi}=0, \text{telos}=8$

$\text{meso}=(\text{arxi}+\text{telos})/2$

το μέσο του πίνακα είναι $(\text{arxi}+\text{telos})/2= 8/2=4$

Άρα $\text{thesi}=4$ και $B[\text{thesi}]=22$.

Το ζητούμενο (9) είναι ίσο με το $B[\text{thesi}]=22$; Όχι. Το 9 είναι μικρότερο από το 22 . Άρα θα είναι κάπου δεξιά του. Εξετάζω στον υποπίνακα στις θέσεις με αριθμό 5,6,7,8 δηλαδή στα στοιχεία

12 9 4 1

Υπολογίζω τα νέα όρια arxi , telos καθώς και το νέο meso

$\text{arxi}=\text{meso}+1$ $\text{arxi}=4+1=5$

$\text{telos}=8$

Βρίσκω το νέο μέσο . $(5+8)/2 \Rightarrow \text{thesi}=6$. Το $B[\text{thesi}]=9$. Το ζητούμενο 9 είναι ίσο με το $B[\text{thesi}]$; Ναι . Άρα το βρήκα στη θέση με αριθμό thesi .

binary_search.c

```
1 /*Binary se fthinon pinaka*/
2 #include<stdio.h>
3 void main(){
4 int zitoumenos, meso,arxi=0,telos=8,thesi,found=0;
5 int B[9]={100,88,75,43,22,12,9,4,1};
6 printf("Pliktrologise zitoymeno\n");
7 scanf("%d",&zitoumenos);
8 if (zitoumenos>B[0]||zitoumenos<B[8])//den exei nohma na anazito ektos orion
9 {
10 printf("Anazitisi adynath\n");
11 }
12 else
13 {
14     while ((!found) && (arxi<=telos))
15     {
16         meso =(arxi+telos)/2;
17         if (B[meso]==zitoumenos)
18         {
19             found=1;
20             thesi=meso;
21         }
22         else if (B[meso]>zitoumenos)
23         {
24             arxi=meso+1;
25         }
26         else
27         {
28             telos=meso-1;
29         }
30     }
31     if (found)
32     {
33         printf("Vrethike sth thesi=%d",thesi);
34     }
35     else
36     {
37         printf("Den Vrethike");
38     }
39 }//telos tou exo else
40 }//end of program..
41
```

while(found==0) &&(arxi<=telos)

Αλγόριθμος ταξινόμησης φυσαλίδας (bubble sort) ή αντιμετάθεσης

Ο αλγόριθμος της φυσαλίδας παίρνει έναν πίνακα μη ταξινομημένο και τον ταξινομεί σε φθίνουσα ή σε αύξουσα σειρά. Υπάρχουν διάφορες υλοποιήσεις που η κάθε μία είναι κοντινή (ή βελτίωση) της άλλης. Όλες στηρίζονται στην εξής γενική λογική:

Έστω πίνακας $\text{int } C[6]=\{3,5,2,1,0,4\}$ και θέλω να τον ταξινομήσω κατά αύξουσα σειρά δηλαδή να μετατραπεί στο τέλος σε $0,1,2,3,4,5$. Ο αλγόριθμος ξεκινάει από το 1ο στοιχείο και το ελέγχει με το 2^ο. Αν είναι μεγαλύτερο τα αντιμεταθέτει, αλλιώς όχι. Κάνει το ίδιο με το 2^ο-3^ο κτλ. Έτσι όταν φτάσει 1 φορά στο τέλος του πίνακα έβγαλε το πρώτο μεγαλύτερο δεξιά. Ξαναπάει στην αρχή και κάνει το ίδιο μέχρι οι διαδοχικές συγκρίσεις να μην απαιτούν άλλες αντιμεταθέσεις. Μέχρι δηλαδή να ταξινομηθεί. Δηλαδή όσο θα είναι αταξινόμητος η διαδικασία επαναλαμβάνεται. Τα πιο πάνω γίνονται πρακτικά ως εξής:

3 5 2 1 0 4

thesi=0 Το $C[\text{thesi}]$ (δηλαδή το 3) είναι μεγαλύτερο από το $C[\text{thesi}+1]$ (5); -όχι .
Μην αντιμεταθέτεις. Προχώρα

thesi=1 Το $C[\text{thesi}]$ (δηλ.5) είναι μεγαλύτερο από το $C[\text{thesi}+1]$ (2); -ναι
αντιμετάθεσε. Άρα ο πίνακας γίνεται:

3 2 5 1 0 4

thesi=2. Το $C[\text{thesi}]$ (δηλ.5) είναι μεγαλύτερο από το $C[\text{thesi}+1]$ (1); -ναι
αντιμετάθεσε. Άρα ο πίνακας γίνεται:

3 2 1 5 0 4

thesi=3. Το $C[\text{thesi}]$ (δηλ.5) είναι μεγαλύτερο από το $C[\text{thesi}+1]$ (0); -ναι
αντιμετάθεσε. Άρα ο πίνακας γίνεται:

3 2 1 0 5 4

thesi=4. Το $C[\text{thesi}]$ (δηλ.5) είναι μεγαλύτερο από το $C[\text{thesi}+1]$ (4); -ναι
αντιμετάθεσε. Άρα ο πίνακας γίνεται:

3 2 1 0 4 5

Δεν πάω στη thesi=5 γιατί δεν υπάρχει το $C[\text{thesi}+1]$ για να συγκρίνω. Ως εδώ έβγαλα το πρώτο μεγαλύτερο του πίνακα (5) στο τέλος. Θα κάνω από την αρχή το ίδιο οπότε κάθε φορά θα βγάζω τον επόμενο μεγαλύτερο προς τα δεξιά..

Πώς δηλαδή;

3 2 1 0 4 5

thesi=0 Το $C[\text{thesi}]$ (δηλαδή το 3) είναι μεγαλύτερο από το $C[\text{thesi}+1]$ (2); -ναι .
Αντιμετάθεσε και προχώρα. . Άρα ο πίνακας γίνεται:

2 3 1 0 4 5

thesi=1 Το $C[\text{thesi}]$ (δηλ.3) είναι μεγαλύτερο από το $C[\text{thesi}+1]$ (1); -ναι
αντιμετάθεσε. Άρα ο πίνακας γίνεται:

2 1 3 0 4 5

thesi=2. Το $C[thesi]$ (δηλ.3) είναι μεγαλύτερο από το $C[thesi+1]$ (0); -ναι
αντιμετάθεσε. Άρα ο πίνακας γίνεται:

2 1 0 3 4 5

thesi=3. Το $C[thesi]$ (δηλ.3) είναι μεγαλύτερο από το $C[thesi+1]$ (4); -όχι . Μην
αντιμεταθέτεις.

Thesi=4. Το $C[thesi]$ (δηλ.4) είναι μεγαλύτερο από το $C[thesi+1]$ (5); -όχι . Μην
αντιμεταθέτεις.

Άρα ο πίνακας γίνεται:

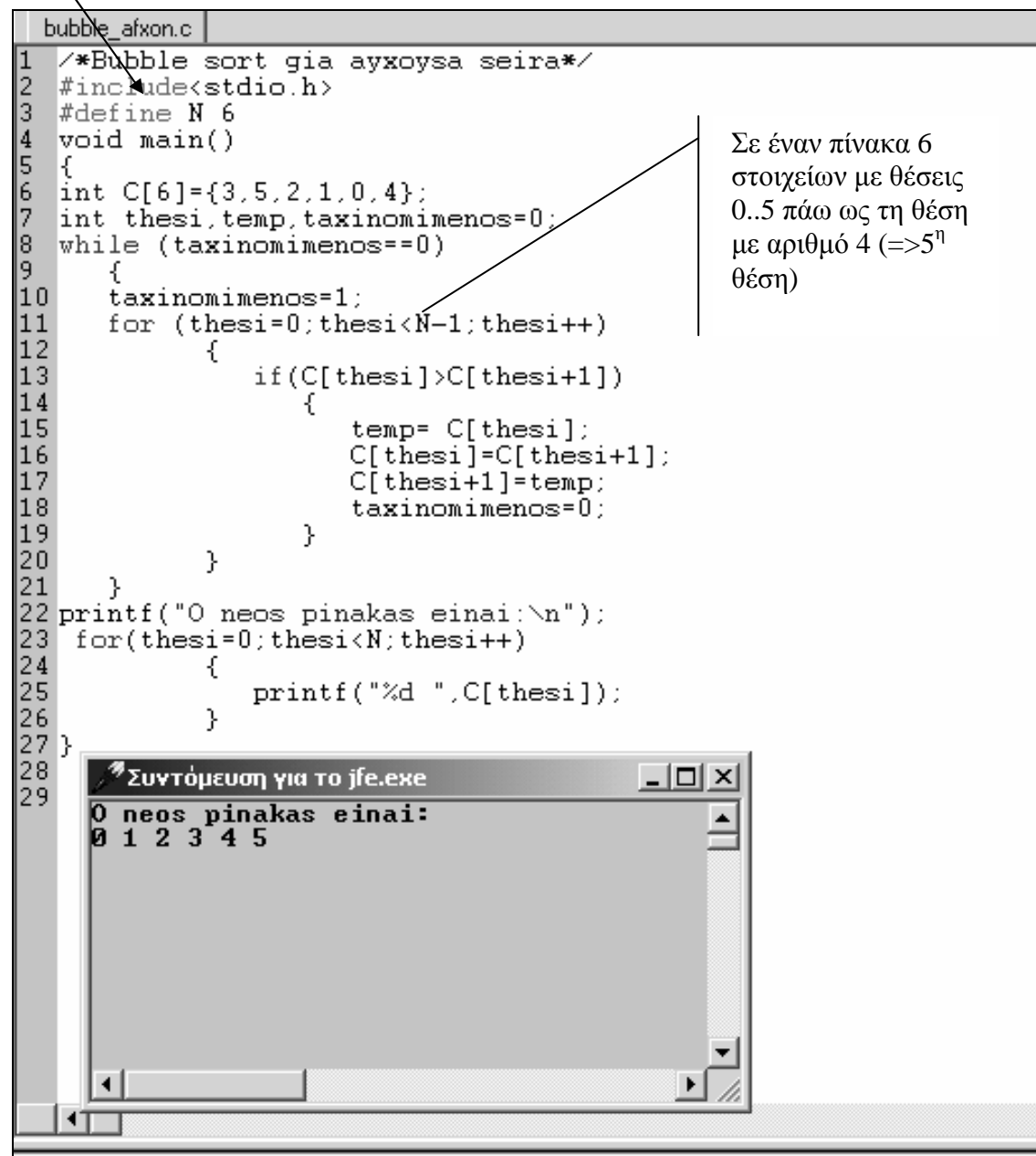
2 1 0 3 4 5

Ο κύκλος αυτός από $thesi=0$ ως $thesi \leq 4$ γίνεται και οι συγκρίσεις με το επόμενο
καταφέρνουν να ταξινομήσουν τον πίνακα. Και να γίνει

0 1 2 3 4 5.

Κάποια στιγμή το πρόγραμμα κάνει συγκρίσεις , αλλά καμία δεν είναι αληθής
γιατί ο πίνακας είναι ταξινομημένος τότε με τη βοήθεια μιας μεταβλητής
σταματάω την έξω επανάληψη όπως θα δω στο παράδειγμα. Παρακάτω.

N είναι το μήκος του πίνακα. Στο προηγούμενο παράδειγμα το N=6 (0..5)



The image shows a C program titled 'bubble_afxon.c' and its execution output. The program implements a bubble sort algorithm. It defines a constant N=6 and an array C with values {3, 5, 2, 1, 0, 4}. A while loop with condition 'taxinomimenos==0' controls the sorting process. Inside, a for loop iterates from thesi=0 to thesi<N-1. A nested if statement compares adjacent elements and swaps them if they are out of order. The variable 'taxinomimenos' is set to 1 when a swap occurs. After the while loop, the sorted array is printed. The execution window shows the output: '0 neos pinakas einai:' followed by the sorted array '0 1 2 3 4 5'.

```
1 /*Bubble sort gia ayxoysa seira*/
2 #include<stdio.h>
3 #define N 6
4 void main()
5 {
6     int C[6]={3,5,2,1,0,4};
7     int thesi,temp,taxinomimenos=0;
8     while (taxinomimenos==0)
9     {
10         taxinomimenos=1;
11         for (thesi=0;thesi<N-1;thesi++)
12         {
13             if(C[thesi]>C[thesi+1])
14             {
15                 temp= C[thesi];
16                 C[thesi]=C[thesi+1];
17                 C[thesi+1]=temp;
18                 taxinomimenos=0;
19             }
20         }
21     }
22     printf("0 neos pinakas einai:\n");
23     for(thesi=0;thesi<N;thesi++)
24     {
25         printf("%d ",C[thesi]);
26     }
27 }
28
29
```

Σε έναν πίνακα 6
στοιχείων με θέσεις
0..5 πάω ως τη θέση
με αριθμό 4 (=>5^η
θέση)

Συντόμευση για το jfe.exe

0 neos pinakas einai:
0 1 2 3 4 5

Η μεταβλητή taxinomimenos είναι αυτή που θα σταματήσει τη while. Πότε;
-Όταν η τιμή της μεταβλητής taxinomimenos γίνει =1 σημαίνει ότι έχω
ταξινομήσει τον πίνακα. Τότε η συνθήκη while(taxinomimenos==0) γίνεται
ψευδής δηλαδή 0 και σταματάει να κάνει επαναλήψεις.

Πώς θα αλλάζαμε το παραπάνω πρόγραμμα να ταξινομεί σε φθίνουσα σειρά;
Πώς θα μετράγαμε στο πιο πάνω πρόγραμμα τις συγκρίσεις που κάνει;
Πώς θα μετράγαμε στο πιο πάνω πρόγραμμα τις αντιμεταθέσεις που κάνει;

Άσκησης

1/5.ΝΓΠ που να γεμίζει έναν πίνακα 5 θέσεων με ακραίους που θα δίνει ο χρήστης από το πληκτρολόγιο και στη συνέχεια να εμφανίζει τα στοιχεία του πίνακα Στη συνέχεια ο χρήστης να πληκτρολογεί έναν αριθμό προς αναζήτηση και το πρόγραμμα με τη βοήθεια της σειριακής –ακολουθιακής αναζήτησης να αναζητά και να εμφανίζει τη θέση /τις θέσεις που βρίσκεται το στοιχείο. Αν το στοιχείο δεν υπάρχει να εμφανίζεται το ανάλογο μήνυμα

2/5.ΝΓΠ που να παίρνει τον πίνακα $\text{int } D[5]=\{3,5,2,1,0,4\}$ να τον ταξινομεί σε φθίνουσα σειρά με τη βοήθεια του αλγορίθμου της φυσαλίδας και στη συνέχεια με τη βοήθεια της διχοτομικής –δυαδικής αναζήτησης να αναζητά την τιμή 0 και να εμφανίζει τη θέση που βρέθηκε.