

Διαδικαστικός Προγραμματισμός

Βασίλης Παλιουράς

Μέχρι τώρα

- Οργάνωση προγράμματος στη C
 - συναρτήσεις
- Μεθοδολογίες σχεδιασμού προγραμμάτων
 - Top-down
 - Λεκτική περιγραφή (προστακτικός τρόπος)
 - Αυξητική ανάπτυξη προγράμματος
- Αναγνωριστές, Τελεστές, Εκφράσεις, Προτάσεις
- Βρόχοι επανάληψης στη C
 - `do { σύνθετη εντολή; } while (έκφραση) ;`
 - `while (έκφραση) { σύνθετη εντολή;}`
 - `for (έκφραση1; έκφραση2; έκφραση3)
 { σύνθετη εντολή }`

Πολλαπλές επιλογές: switch

```
switch (έκφραση) {  
  case τιμή1: εντολές ; break;   
  case τιμή2: εντολές ; break;   
  case τιμή3: εντολές ; break;   
  /* ... */  
  default: εντολές ; break;  
}
```

case τιμή1: λειτουργεί ως

label

break: μεταφέρει τον έλεγχο

εκτός του switch () {}

τι γίνεται χωρίς **break**

```
#include <stdio.h>
int getchoice (void) ;
void start (void) ;
void stop (void);

int main (){

    int userchoice ;

    while ((userchoice = getchoice()) != 3){
        switch (userchoice) {
            case 1: start() ;
                    break;
            case 2: stop();
                    break;
            default: break;
        }
    }

    return 0;
}
```

```
int getchoice (void ) {
    int choice ;

    printf("1: start\n2: stop\n3:quit\n");
    printf("enter choice:\n");
    scanf("%d", &choice);

    return choice;
}

void start (void) {
    printf("Start...");
}

void stop (void) {
    printf("Stop...");
}
```

Πίνακες στη C

- Δεσμεύουν **συνεχή χώρο** στη μνήμη
- Οι ακόλουθες δηλώσεις οδηγούν τον compiler να δημιουργήσει διαφορετική assembly.

```
int a[32];
```

```
int s = 32;
```

```
int a[s];
```

Παράδειγμα

```
#include <stdio.h>
```

```
int main ( ) {
```

```
    int i;
```

```
    float temp[3] = { -2.1, 5.5, 10.1};
```

```
    for (i=0; i< 3; i = i + 1)
```

```
        printf("\ttemp[%d]: %f\n", i, temp[i]);
```

```
    return 0;
```

```
        printf("\ttemp[%d]: %+6.2f\n", i, temp[i]);
```

[cygwin](#)

```
}
```

Πίνακες

- Συλλογή μεταβλητών **ίδιου τύπου**, οι οποίες αποθηκεύονται σε διαδοχικές θέσεις μνήμης.
- **float temperature[31];**
 - δήλωση πίνακα μεταβλητών **float**, 31 στοιχείων
 - temperature[**0**] είναι το **πρώτο** στοιχείο,
 - temperature[**1**] είναι το **δεύτερο** στοιχείο,
 - ...
 - temperature[**30**] είναι το **τριακοστό πρώτο** στοιχείο,
 - temperature είναι **η διεύθυνση του πρώτου στοιχείου**
 - temperature είναι το ίδιο με &temperature[0]

```
#include <stdio.h>
#define N 10
```

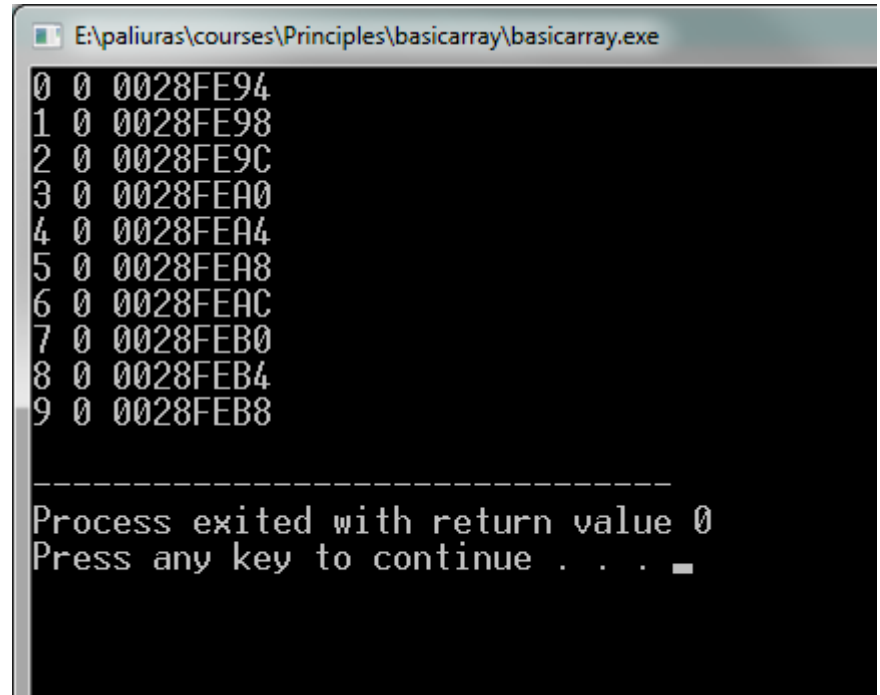
```
int main() {
```

```
    int i ;
    int data[N] = {0};
```

```
    for (i=0; i< 10; i++)
        printf("%d %d %p\n", i, data[i], &data[i]);
```

```
    return 0;
```

```
}
```



```
E:\paliuras\courses\Principles\basicarray\basicarray.exe
0 0 0028FE94
1 0 0028FE98
2 0 0028FE9C
3 0 0028FEA0
4 0 0028FEA4
5 0 0028FEA8
6 0 0028FEAC
7 0 0028FEB0
8 0 0028FEB4
9 0 0028FEB8

-----
Process exited with return value 0
Press any key to continue . . .
```

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

Πίνακες δύο (ή περισσότερων) διαστάσεων

- `int a[3][3] ;`
- `int a[3][3] = {{1,2,3}, {3,2,1}, {1,1,1}};`

1	2	3
3	2	1
1	1	1

```
#include <stdio.h>
#define N 3

int main ( ) {
int i, j;

int a[N][N] = {{1,2,3}, {3,2,1}, {1,1,1}};

for (i=0; i<N; i++) {
    for (j=0; j<N; j++)
        printf("%d ",a[i][j]);
    printf("\n");
}

return 0;
}
```

Αποθήκευση στη μνήμη - πίνακας μιας διάστασης

```
int a[3] = {10, 20, 30};
```

10	20	30
----	----	----

Διεύθυνση	Στοιχείο	Περιεχόμενα
1000	a[0]	10
1004	a[1]	20
1008	a[2]	30

Αποθήκευση στη μνήμη πίνακας δύο διαστάσεων

a[1] σημαίνει δεύτερη γραμμή
a[1][2] σημαίνει τρίτο στοιχείο
δεύτερης γραμμής

1	2	3
3	2	1
1	1	1

Διεύθυνση	Στοιχείο
1000	1
1004	2
1008	3
100C	3
1010	2
1014	1
1018	1
101C	1
1020	1

```
#include <stdio.h>
#define N 3

int main ( ) {
int i, j;

int a[N][N] = {{1,2,3}, {3,2,1}, {1,1,1}};
int *b = &a[0][0];

for (i=0; i< N; i++) {
    for (j=0; j< N; j++)
        printf("%d ",a[i][j]);
    printf("\n");
}

for (i=0; i< N*N; i++)
    printf("%d ", *(b+i));

system("pause");
return 0;
}
```

Συνάρτηση της βασικής βιβλιοθήκης scanf ()

```
int number;
```

```
char ch;
```

%d θα διαβάσει ακέραιο

```
scanf("%d", &number);
```

%c θα διαβάσει χαρακτήρα

```
scanf("%c", &ch);
```

τελεστής διεύθυνσης (&):

Επιστρέφει τη διεύθυνση

της θέσης μνήμης η οποία

αντιστοιχεί στη μεταβλητή

που ακολουθεί

Παράδειγμα

```
#define N 2
#include <stdio.h>
int main ( ) {
    int data[N][N] ;
    int i, j ;

    for (i =0 ; i < N ; i++)
        for ( j = 0 ; j < N ; j ++ ) {
            printf ("element (%d,%d)?\t", i, j);
            scanf("%d", &data[i][j]);
        }

    for (i =0 ; i < N ; i++) {
        for ( j = 0 ; j < N ; j ++ )
            printf ("%d\t", data[i][j]);
        printf("\n");
    }
    return 0;
}
```

```

#define N 2
#include <stdio.h>
void readdata(int [N][N]);
void writedata(int [N][N]);

int main ( ) {
    int data[N][N] ;

    readdata(data) ;
    writedata(data);

    return 0;
}

```

```

void readdata(int a[N][N]) {
    int i,j;
    for (i =0 ; i < N ; i++)
        for ( j = 0 ; j < N ; j ++ ) {
            printf ("element (%d,%d)?\t", i, j);
            scanf("%d", &a[i][j]);
        }
}

```

```

void writedata(int b[N][N]) {
    int i,j;
    for (i =0 ; i < N ; i++) {
        for ( j = 0 ; j < N ; j ++ )
            printf ("%d\t", b[i][j]);
        printf("\n");
    }
}

```

Καλύτερα!

ΠΡΟΣΟΧΗ!!! ΤΕΡΑΣΤΙΟ ΛΑΘΟΣ!!!

Δεν χρησιμοποιείται ο πίνακας data αλλά περιοχή μνήμης που αρχίζει στη διεύθυνση που περιέχεται στο data[N][N], το οποίο είναι εκτός του πίνακα.

```
example5 - [example5bad.dev] - Dev-C++ 5.4.1
File Edit Search View Project Execute Debug Tools CVS Window Help
(globals)
Project Classes Debug
example5
  example5main.c
  readdata.c
  writedata.c
1 #include <stdio.h>
2 #define N 2
3 void readdata (int [N][N]);
4 void writedata(int [N][N]);
5
6 int main ( ) {
7     int data[N][N] ;
8
9     readdata(data[N][N]);
10    writedata(data[N][N]);
11
Close
principles\1516\lecture06\example5bad\example5main.c
principles\1516\lecture06\example5bad\
principles\1516\lecture06\example5bad\
argument 1 of 'readdata' makes pointer from integer without a cast [enabled by default]
*)[2]' but argument is of type 'int'
[Warning] passing argument 1 of 'writedata' makes pointer from integer without a cast [enabled by default]
E:\paliuras\courses\Principles\1516\lecture... [Note] expected 'int (*)[2]' but argument is of type 'int'
Line: 9 Col: 25 Sel: 0 Lines: 13 Insert Done parsing
lecture06 Microsoft Po... example5 - D... EN (1:33) 10:48 μμ 7/3/2016
```

Λόγω του λάθους αυτού, το Πρόγραμμα μπορεί να έχει απρόβλεπτη συμπεριφορά. Σε μερικές περιπτώσεις μπορεί να φαίνεται ότι λειτουργεί, αλλά θα δημιουργήσει προβλήματα μόλις Προσθεθεί περαιτέρω κώδικας.

Ο compiler δίνει warnings

```
#define N 2
#include <stdio.h>
void readdata(int a[N][N]);
void writedata(int b[N][N]);
int sumdata(int x[N][N]);

int main ( ) {
    int data[N][N] ;

    readdata(data) ;
    writedata(data);

    printf("The sum is: %d\n",
    sumdata(data));

return 0;
}
```

```
int sumdata(int x[N][N]) {
    int i, j;
    int sum = 0;

    for (i=0; i<N; i++)
        for (j=0; j<N; j++)
            sum += x[i][j];

return sum;
}
```

Στη μνήμη υπάρχει
μόνο ένας πίνακας!

- Συνάρτηση main

- Πίνακας data
- Καλείται η συνάρτηση readdata
 - Όρισμα a
- Καλείται η συνάρτηση writedata
 - Όρισμα b

Πίνακας
δεδομένων data

```
#define N 2
#include <stdio.h>
void readdata(int a[N][N]);
void writedata(int b[N][N]);
int sumdata(int x[N][N]);

int main ( ) {
    int data[N][N] ;

    readdata(data) ;
    writedata(data);

    printf("The sum is: %d\n",
           sumdata(data));

    return 0;
}
```

Παράδειγμα

- Ζητήθηκε από 40 φοιτητές να βαθμολογήσουν το φαγητό στο κυλικείο από 1 (απαίσιο) έως και 10 (αστέρι michelin).
- Να συνοψίσουμε τα αποτελέσματα.

- Από Deitel & Deitel.

- Λεκτική περιγραφή – Προστακτικός προγραμματισμός: (οργάνωση προγράμματος – συναρτήσεις)

Διάβασε τις απαντήσεις

Υπολόγισε το ιστόγραμμα \Rightarrow

`generatehist()`

Παρουσίασε το αποτέλεσμα \Rightarrow `printhist()`

- Αναπαράσταση δεδομένων

- Πίνακας για απαντήσεις \Rightarrow `responses[]`

- Πίνακας για ιστόγραμμα \Rightarrow `frequency[]`

Πρότυπα συναρτήσεων



```
#include <stdio.h>
#include <stdlib.h>

#define RESPONSE_SIZE 40
#define FREQUENCY_SIZE 11

void generatehist(int [], int []);
void printhist(int []);
void printstars(int );

int main()
{
    int frequency[ FREQUENCY_SIZE ] = { 0 };
    int responses[ RESPONSE_SIZE ] =
        { 1, 2, 6, 4, 8, 5, 9, 7, 8, 10,
          1, 6, 3, 8, 6, 10, 3, 8, 2, 7,
          6, 5, 7, 6, 8, 6, 7, 5, 6, 6,
          5, 6, 7, 5, 6, 4, 8, 6, 8, 10 };

    generatehist(frequency, responses);

    printhist(frequency);

    return 0;
}
```



Υλοποίηση συναρτήσεων



```
void generatehist(int frequency[], int responses[]){
    int answer;

    for ( answer = 0; answer <= RESPONSE_SIZE - 1; answer++ )
        ++frequency[ responses [ answer ] ];
}

void printhist(int frequency[]) {
    int rating;

    printf( "%s%17s Bar\n", "Rating", "Frequency" );
    for ( rating = 1; rating <= FREQUENCY_SIZE - 1; rating++ ) {
        printf( "%6d%17d ", rating, frequency[ rating ] );
        printstars(frequency[rating]);
        printf("\n");
    }
}

void printstars(int len){
    int i;

    for (i=0;i<len;i++)
        putchar('*');
}
```

Η λεκτική περιγραφή αντιστοιχίζεται σε κλήση συναρτήσεων.

```

#include <stdio.h>
#define RESPONSE_SIZE 40
#define FREQUENCY_SIZE 11

int main()
{
    int answer, rating, frequency[ FREQUENCY_SIZE ] = { 0 };
    int responses[ RESPONSE_SIZE ] =
        { 1, 2, 6, 4, 8, 5, 9, 7, 8, 10,
          1, 6, 3, 8, 6, 10, 3, 8, 2, 7,
          6, 5, 7, 6, 8, 6, 7, 5, 6, 6,
          5, 6, 7, 5, 6, 4, 8, 6, 8, 10 };

    for ( answer = 0; answer <= RESPONSE_SIZE - 1; answer++ )
        ++frequency[ responses [ answer ] ];

    printf( "%s%17s\n", "Rating", "Frequency" );

    for ( rating = 1; rating <= FREQUENCY_SIZE - 1; rating++ )
        printf( "%6d%17d\n", rating, frequency[ rating ] );

    return 0;
}

```