



«Προγραμματισμός Ι»

5- ΔΙΑΤΑΞΕΙΣ & ΑΛΦΑΡΙΘΜΗΤΙΚΑ

Δρ. Χαράλαμπος Καρανίκας
Τμήμα Μηχανικών Πληροφορικής

ΔΙΑΤΑΞΕΙΣ

- Δήλωση μονοδιάστατων διατάξεων (arrays)
- Λίστα μεταβλητών **ίδιου τύπου** που προσπελάζονται από **κοινό όνομα**
- Μεμονωμένη μεταβλητή ~ στοιχείο της διάταξης
- τύπος όνομα[μέγεθος];

```
int myarray[20];  
myarray[1]  
myarray[0] = 100;
```

```
int i[5];  
int j;
```

```
for(j=0; j<5; j++) i[j] = j;
```

```
int sqrs[10];  
int i;
```

```
for(i=1; i<11; i++) sqrs[i-1] = i*i;  
for(i=0; i<10; i++) printf("%d ", sqrs[i]);
```

```
scanf("%d", &count[9]);
```

```
/*Για αντιγραφή κάθε στοιχείο ξεχωριστά*/
```

```
char a1[10], a2[10];
```

```
...
```

```
a2 = a1; /*Λάθος*/
```

Παραδείγματα I

```
int main(void)
{
    int temp[31], i, min, max, avg;
    int days;

    printf("How many days in the month? ");
    scanf("%d", &days);

    for(i=0; i<days; i++) {
        printf("Enter noontday temperature for day %d: ", i+1);
        scanf("%d", &temp[i]); }

    /*Μέσος όρος*/
    avg = 0;
    for(i=0; i<days; i++) avg = avg + temp[i];
    printf("Average temperature: %d\n", avg/days);

    /*Ελάχιστο/Μέγιστο*/
    min = 200; /* initialize min and max */
    max = 0;
    for(i=0; i<days; i++) {
        if(min>temp[i]) min = temp[i];
        if(max<temp[i]) max = temp[i]; }

    printf("Mini/Max temperature: %d\n %d\n", min, max);
    return 0;
}
```

```
#include <stdio.h>

int main(void)
{
    int a1[10], a2[10];
    int i;

    for(i=1; i<11; i++) a1[i-1] = i;

    for(i=0; i<10; i++) a2[i] = a1[i];

    for(i=0; i<10; i++)
        printf("%d ", a2[i]);

    return 0;
}

/*Άσκηση: Αντιγραφή και
αντεστραμμένα*/
```

Παραδείγματα II

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

int main(void)
{
    char mess[80];
    int i;

    printf("Enter msg (< 80 characters)\n");
    for(i=0; i<80; i++) {
        mess[i] = getche();
        if(mess[i]=='\r') break;
    }
    printf("\n");

    for(i=0; mess[i]!='\r'; i++)
        printf("%c", mess[i]+1);

    return 0;
}
```

```
#include <stdio.h>
int main(void){
    int item[100];
    int a, b, t, count;

    printf("How many numbers? ");
    scanf("%d", &count);

    for(a=0; a<count; a++) scanf("%d", &item[a]);

    /* Ταξινόμηση των αριθμών με bubble sort */
    for(a=1; a<count; ++a)
        for(b=count-1; b>=a; --b) {
            /* Σύγκριση γειτονικών στοιχείων */
            if (item[b-1] > item[b]) {
                /* Εναλλαγή στοιχείων */
                t = item[b-1];
                item[b-1] = item[b];
                item[b] = t;
            }
        }
    for(t=0; t<count; t++) printf("%d ", item[t]);

    return 0;
}
```

ΧΡΗΣΗ ΑΛΦΑΡΙΘΜΗΤΙΚΩΝ

- Αλφαριθμητικό ~ Διάταξη χαρακτήρων τερματιζόμενη με κενό (null)
- Στην C το κενό είναι το μηδέν. Άρα μέγεθος διάταξης = 1 byte μεγαλύτερο από το μέγεθος του αλφαριθμητικού ώστε να χωράει ο χαρακτήρας null.

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

```
int main(void)
```

```
{
```

```
    char str[80];
```

```
    int i;
```

```
    printf("Enter a string (less than 80 chars):");
```

```
    gets(str);
```

```
    /* 1ος τρόπος παρουσίασης */
```

```
    for(i=0; str[i]; i++) printf("%c", str[i]);
```

```
    /* 2ος τρόπος παρουσίασης */
```

```
    printf(str); /* output the string */
```

```
    /* 3ος τρόπος παρουσίασης */
```

```
    printf("%s\n", str);
```

```
    return 0;
```

```
}
```

strcpy (προορισμός, προέλευση);

```
#include <string.h>
```

```
    char str[80];
```

```
    strcpy(str, "hello");
```

```
    printf("%s", str);
```

strcat (προορισμός, προέλευση);

```
    char str[80];
```

```
    strcpy(str, "hello");
```

```
    strcat(str, " there");
```

```
    printf(str);
```

strcmp (s1, s2);

```
    printf("%d", strcmp("one", "one"));
```

strlen (αλφαριθμητικό);

```
    printf("%d", strlen("test"));
```

Παραδείγμα Ι

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

int main(void)
{
    char str1[80], str2[80];
    int i;

    printf("Enter the first string: "); gets(str1);
    printf("Enter the second string: "); gets(str2);

    printf("%s is %d chars long\n", str1, strlen(str1));
    printf("%s is %d chars long\n", str2, strlen(str2));

    i = strcmp(str1, str2);
    if(!i) printf("The strings are equal.\n");
    else if(i<0) printf("%s is less than %s\n", str1, str2);
    else printf("%s is greater than %s\n", str1, str2);

    if(strlen(str1) + strlen(str2) < 80) {
        strcat(str1, str2);
        printf("%s\n", str1);
    }

    strcpy(str1, str2);
    printf("%s %s\n", str1, str2);

    return 0;
}
```

Παραδείγμα II

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
int main(void)
{ char command[80], temp[80];
  int i, j;

  for( ; ; ) {
    printf("Operation? ");   gets(command);

    if(!strcmp(command, "quit")) break;

    printf("Enter first number: ");   gets(temp);
    i = atoi(temp);
    printf("Enter second number: ");   gets(temp);
    j = atoi(temp);

    if(!strcmp(command, "add"))
      printf("%d\n", i+j);
    else if(!strcmp(command, "subtract"))
      printf("%d\n", i-j);
    else if(!strcmp(command, "divide")) {
      if(j) printf("%d\n", i/j);
    }
    else if(!strcmp(command, "multiply"))
      printf("%d\n", i*j);
    else printf("Unknown command. \n");
  }
  return 0;
}
```

ΔΗΜΙΟΥΡΓΙΑ ΠΟΛΥΔΙΑΣΤΑΤΩΝ ΔΙΑΤΑΞΕΩΝ -1

- Πολυδιάστατες διατάξεις. Καθορίζονται με το μέγεθος των επιπλέον διαστάσεων μέσα σε αγκύλες.
 - πχ. Για ένα δισδιάστατο πίνακα έχουμε `int A[2][5]`; (2 γραμμές και 5 στήλες)
 - πχ. `int B[2][3][2]`
 - Η προσπέλαση γίνεται γραμμή προς γραμμή από αριστερά προς τα δεξιά.
- Μία δισδιάστατη διάταξη είναι ουσιαστικά μία διάταξη μονοδιάστατων διατάξεων.
- Μία τρισδιάστατη διάταξη είναι μία διάταξη δισδιάστατων διατάξεων.

ΔΗΜΙΟΥΡΓΙΑ ΠΟΛΥΔΙΑΣΤΑΤΩΝ ΔΙΑΤΑΞΕΩΝ-2

```
#include <stdio.h>

int main(void)
{
    int twod[4][5];
    int i, j;

    for(i=0; i<4; i++)
        for(j=0; j<5; j++)
            twod[i][j] = i*j;

    for(i=0; i<4; i++) {
        for(j=0; j<5; j++)
            printf("%d ", twod[i][j]);
        printf("\n");
    }

    return 0;
}
```

```
#include <stdio.h>
/* Σκόρ κάθε μέλους ομάδας μπάσκετ ανά τρίμηνο*/
int main(void)
{
    int bball[4][5];
    int i, j;

    for(i=0; i<4; i++)
        for(j=0; j<5; j++) {
            printf("Quarter %d, player %d, ", i+1, j+1);
            printf("Enter number of points: ");
            scanf("%d", &bball[i][j]);
        }

    for(i=0; i<4; i++)
        for(j=0; j<5; j++) {
            printf("Quarter %d, player %d, ", i+1, j+1);
            printf("%d\n", bball[i][j]);
        }

    return 0;
}
```

ΑΡΧΙΚΟΠΟΙΗΣΗ ΔΙΑΤΑΞΕΩΝ

- τύπος όνομα-διάταξης[μέγεθος] = {λίστα-τιμών};

- `int i[5] = {1, 4, 9, 16, 25}; /* i[0]=1, i[4]=25*/`

Διατάξεις χαρακτήρων με δύο τρόπους:

- `char a[3] = {'A', 'B', 'C'}; /*Δεν τερματίζει με null */`

- `char name[5] = "Herb"; /*Χωρίς άγκιστρα και το null μπαίνει αυτόματα*/`

- `int sqr[3][3] = {`

- `1, 2, 3,`

- `4, 5, 6,`

- `7, 8, 9`

- `};`

- `int pwr[ ] = {1, 2, 4, 8, 16, 32, 64, 128}; /*Χωρίς μέγεθος*/`

`char prompt[] = "Enter your name: ";`

`int sqr[][3] = {`

`1, 2, 3,`

`4, 5, 6,`

`7, 8, 9`

`};`

Παραδείγματα

```
#include <stdio.h>

int main(void)
{
    int ServerUsers[5][2] = {
        1, 14,
        2, 28,
        3, 19,
        4, 8,
        5, 15
    };

    int server;
    int i;

    printf("Enter the server number: ");
    scanf("%d", &server);

    /* look it up in the table */
    for(i=0; i<5; i++)
        if(server == ServerUsers[i][0]) {
            printf("There are %d users on server %d.\n",
                ServerUsers[i][1], server);
            break;
        }

    /* report error if not found */
    if(i==5) printf("Server not listed.\n");

    return 0;
}
```

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

int main(void)
{
    char str[80] = "I like C";

    strcpy(str, "hello");
    printf(str);

    return 0;
}
```

ΚΑΤΑΣΚΕΥΗ ΔΙΑΤΑΞΕΩΝ ΑΛΦΑΡΙΘΜΗΤΙΚΩΝ - 1

- `char names[10][40];`

Ένας πίνακας με 10 αλφαριθμητικά, καθένα με 40 χαρακτήρες(με null).

Γιά προσπέλαση καθορίζουμε μόνο τον αριστερότερο δείκτη στοιχείου.

πχ. Για το 3^ο αλφαριθμητικό `names[2]`, για το 1^ο
`printf(names[0])`

ΚΑΤΑΣΚΕΥΗ ΔΙΑΤΑΞΕΩΝ ΑΛΦΑΡΙΘΜΗΤΙΚΩΝ - 2

```
#include <stdio.h>

int main(void)
{
    char text[10][80];
    int i;

    for(i=0; i<10; i++) {
        printf("%d: ", i+1);
        gets(text[i]);
    }

    do {
        printf("Enter number of string (1-10) : ");
        scanf("%d", &i);
        i--; /* adjust value to match array index */
        if(i>=0 && i<10) printf("%s\n", text[i]);
    } while(i>=0);

    return 0;
}
```

```
char words[][2][40] = {
    "dog", "Hund",
    "no", "nein",
    "year", "Jahr",
    "child", "Kind",
    "I", "Ich",
    "drive", "fahren",
    "house", "Haus",
    "to", "zu",
    "", ""
};

int main(void)
{
    char english[80];
    int i;

    printf("Enter English word: ");
    gets(english);
    i = 0;
    while(strcmp(words[i][0], "")) {
        if(!strcmp(english, words[i][0])) {
            printf("German translation: %s", words[i][1]);
            break;
        }
        i++;
    }
    if(!strcmp(words[i][0], "")) printf("Not in dictionary\n");
    return 0;
}
```