

Εργαστήριο 12

Δομές (structures).

Εισαγωγικά, περιγραφή δομής, πεδία δομής.

Δηλώσεις δομών και δεδομένα στις δομές.

Φωλιασμένες δομές.

Δομές ως παράμετροι συναρτήσεων

και ως τιμή επιστροφής συναρτήσεων.

Πίνακες δομών.

Παράδειγμα. Μιγαδικοί αριθμοί.

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

struct complex { float re, im; }; // δήλωση νέου τύπου

main(){ struct complex z1, z2; // δηλώσεις μεταβλητών

    printf("sizeof = %d \n", sizeof(struct complex));

    printf("give z1.re -> ");

    scanf( "%f", &z1.re ); // διάβασμα

    z1.im := 55;

    z2 = z1; // εκχώρηση δομών

    printf("z2 = %f + i * %f \n", z2.re, z2.im );

    system("pause");
}
```

Παράδειγμα. Φωλιασμένες δομές.

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

struct color { int r,g,b; }; // red, green, blue

struct pixel {
    int x,y; // συντεταγμένες pixel
    struct color c; // δομή color μέσα σε pixel
};
```

```

main() {

    struct pixel p;
    struct color blue;

    blue.r = blue.g = 0; blue.b = 255; // create blue

    p.x = 12; p.y = 50;

    p.c = blue; // set pixel color to blue

    p.c.r = 50; // correct red of p, but NOT of blue

    printf("x=%d, y=%d, r=%d, g=%d, b=%d \n",
           p.x, p.y, p.c.r, p.c.g, p.c.b );

    system("pause");
}

```

Παράδειγμα. Μιγαδικοί αριθμοί. Συναρτήσεις και Δομές.

```

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

struct complex { float re, im; };

struct complex add ( struct complex a, struct complex b ){

    struct complex help;

    help.re = a.re + b.re;
    help.im = a.im + b.im;

    return help;
}

main(){ struct complex z1, z2, z3; // μεταβλητές

    z1.re=z1.im = 1; // z1=1+i

    z2.re=z2.im = 6; // z2=6+6i

    z3 = add( z1, z2 ); // κλήση add

    printf("z3 = %f + i * %f \n", z3.re, z3.im );

    system("pause");
}

```

Παράδειγμα. Operator overloading στη C++.

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

struct complex { float re, im; };

struct complex operator + ( struct complex a, struct complex b ){

    struct complex help;

    help.re = a.re + b.re;
    help.im = a.im + b.im;

    return help;
}

main(){ struct complex z1, z2, z3;

    z1.re=z1.im = 1; // z1=1+i
    z2.re=z2.im = 6; // z2=6+6i

    z3 = z1 + z2; // overload του +

    printf("z3 = %f + i * %f \n", z3.re, z3.im );

    system("pause");
}
```

Παράδειγμα. Πίνακες δομών.

```
#include<stdio.h>
#include<stdlib.h>
#define N 100

struct student { int am; char name[30]; };

main(){ struct student pin[N]; // πίνακας δομών

    int i;

    for(i=0;i<N;i++){ // διάβασμα
        printf("give AM of student %d -> ", i );
        scanf("%d", &pin[i].am ); // note &
        getchar(); // why getchar here?
        printf("give name -> ");
        gets(pin[i].name);
    }
}
```

```
for(i=0;i<N;i++){ // εκτύπωση
    printf("student %2d, am=%4d, name=%s \n",
           i, pin[i].am, pin[i].name );
}

system("pause");
}
```

Διαγώνισμα T4.