

# STRUCTS-EXTRA

COURSE-5

# ΕΙΣΑΓΩΓΗ

```
//1.Δήλωση ενός struct
typedef struct
{
    /* data */
    int y;
}temp /*->Name_of_struct*/;
//2.Δήλωση ενός struct[εναλλακτικός τρόπος]
struct temp/*-> name_of_struct*/
{
    /* data */
    int y;//Ακέραιο δεδομένο y.
    //Ανήκει στον τύπο δεδομένων temp!!!!
};
int main()
{
    /*1.*/ temp t1;
    /*2.*/ struct temp t2;
    //Διάβασμα τιμών από struct
    scanf("%d",&t1.y);
    scanf("%d",&t2.y);
    temp temps[5];
    //Πίνακας πέντε θέσεων αποτελούμενος από δεδομένα τύπου temp.
    temps[0].y;/*Χρήση δεδομένου y για την μεταβλητή στην θέση 0
    του πίνακα.*/
}
```

# 1.ΜΕΤΑΒΛΗΤΕΣ ΚΑΙ ΔΕΙΚΤΕΣ ΣΕ ΔΟΜΕΣ

```
typedef struct
{
    char brand[100];
    char *model;
    double price;
}car;
//νέος τύπος δεδομένων car.
//3 δεδομένα στον τύπο car.
int main()
{
    //Δέσμευση μνήμης για μεταβλητή τύπου car.
    car *c1=(car *)malloc(sizeof(car));
    //χρήση αριθμητικών δεδομένων από δείκτη(->).
    c1->brand="toyota";
    //!!!Προσοχή το αριθμητικό δεδομένο model
    //είναι δείκτης άρα πρέπει να δεσμεύσω μνήμη για αυτό.
    c1->model=(char *)malloc(100 * sizeof(char));
    //Πλέον μπορώ να χρησιμοποιήσω την μεταβλητή model.
    //Εναλλακτικός τρόπος κλήσης αριθμητικού δεδομένου((*).).
    (*c1).price=19000;
    //Αποδέσμευση μνήμης.
    //πρώτα για την μεταβλητή model Αποδεσμεύω την μνήμη της
    free(c1->model);
    //Αποδέσμευση μνήμης για structs
    free(c1);
}
```

## 2.ΔΟΜΕΣ ΚΑΙ ΣΥΝΑΡΤΗΣΕΙΣ

```
#include <stdio.h>
//Χρήση δεικτών στις δομές
typedef struct
{
    int x;
    int y;
}point;
void sort(point *points,int size)
{
    //ταξινόμηση σημείων με πολλαπλασιασμό
    for(int i=0;i<size;i++)
    {
        for(int j=0;j<size-1;j++)
        {
            if((points[j].x*points[j].y)>(points[j+1].x * points[j+1].y))
            {
                point temp=points[j];
                points[j]=points[j+1];
                points[j+1]=temp;
            }
        }
    }
}
```



## 2.ΔΟΜΕΣ ΚΑΙ ΣΥΝΑΡΤΗΣΕΙΣ[ΣΥΝΕΧΕΙΑ]

```
int main()
{
    //Απευθείας αρχικοποίηση τιμών σε πίνακα με σημεία.
    point points[5]={1,2},{4,5},{1,7},{4.6},{9,8}};
    sort();
    for(int i=0;i<5;i++)
    {
        printf("{%d,%d}\n",points[i].x,points[i].y);
    }
}
```

## 3.ΓΕΝΙΚΑ

- Ένα struct μας επιτρέπει να εισάγουμε custom τύπους στα προγράμματα μας.
- Ένα struct περιέχει παραπάνω από 1 αριθμητικά δεδομένα.
- Για να συγκρίνω μεταβλητές πρέπει να τις συγκρίνω με βάση ένα αριθμητικό δεδομένο.
- Μπορώ να εκχωρήσω ένα struct σε ένα άλλο.
- Απευθείας αρχικοποίηση σε struct.(Π.χ point) point p1={2,5};
- Τα structs όπως και οι υπόλοιποι τύποι δεδομένων μπορούν να χρησιμοποιηθούν σε συναρτήσεις και να δεσμεύσουν δυναμικά μνήμη.
- Κλήση τιμής από κάποιο struct(Π.Χ p1.x).
- Κλήση τιμής από δείκτη struct(Π.Χ p1->x).

# ΠΑΡΑΔΕΙΓΜΑ\_1

- Να κατασκευαστεί μία δομή η οποία θα αναπαριστά εργαζόμενους με αριθμητικά δεδομένα το όνομα του εργαζόμενου, το μισθό του.
- Να κατασκευαστεί συνάρτηση η οποία θα ταξινομεί τους εργαζόμενους με βάση τον μισθό τους.
- Να κατασκευαστεί συνάρτηση η οποία θα αποθηκεύει σε αρχείο τα δεδομένα μετά την ταξινόμηση
- Στην κύρια συνάρτηση να κατασκευαστεί πίνακας για 10 εργαζομένους για τους οποίους οι τιμές θα εισάγονται από τον χρήστη και να καλούνται οι παραπάνω συναρτήσεις.

# ΠΑΡΑΔΕΙΓΜΑ\_1[ΛΥΣΗ]

```
#include <stdio.h> //FILE I/O
#include <stdlib.h> //malloc free
typedef struct      //Δήλωση δομής για αναπαράσταση εργαζομένων
{
    char name[100]; //Αλφαριθμητικό δεδομένο
    double salary;  //Δεδομένο δεκαδικού τύπου.
}employee;
void sort(employee *employees, int size) //Αλγόριθμος ταξινόμησης bubblesort
{
    for(int i=0; i<size; i++)
    {
        for(int j=0; j<size-1; j++)
        {
            if(employees[j].salary > employees[j+1].salary) //Σύγκριση εργαζομένων με βάση τον μισθό τους
            {
                //swap(αντιστροφή εργαζομένων).
                employee temp = employees[j];
                employees[j] = employees[j+1];
                employees[j+1] = temp;
            }
        }
    }
}
```



# ΠΑΡΑΔΕΙΓΜΑ\_1[ΛΥΣΗ]

```
void save_to_file(employee *employees,int size)
{
    FILE *fp;//Δείκτης σε αρχείο.
    char filename[100];
    fflush(stdin);//Άδειασμα buffer εντολών.
    printf("Give filename:");
    gets(filename);//εισαγωγή ονόματος από τον χρήστη.
    fp=fopen(filename,"w");//Άνοιγμα αρχείου για γράψιμο.
    //Αν δεν υπάρχει το αρχείο δημιουργείται.
    for(int i=0;i<size;i++)
    {
        fprintf(fp,"Name:%s \t Salary:%.2lf \n",employees[i].name,employees[i].salary);
        //Εκτύπωση σε αρχείο με το παραπάνω format("Name:%s \t Salary:%.2lf \n").
    }
    fclose(fp);//κλείσιμο αρχείου.
}
```

# ΠΑΡΑΔΕΙΓΜΑ\_1[ΛΥΣΗ]

```
int main()
{
    employee *employees;
    //Δείκτης σε μεταβλητή/τες τύπου employee
    employees=(employee *)malloc(5 * sizeof(employee));
    //Δέσμευση 5 θέσεων για τον δείκτη employee
    //Θα δείχνει σε πίνακα.
    for(int i=0;i<5;i++)
    {
        //Διάβασμα δεδομένων από τον χρήστη.
        fflush(stdin);
        printf("Give name of employee:");
        gets(employees[i].name);
        printf("Give salary of employee:");
        scanf("%lf",&employees[i].salary);
    }
    sort(employees,5);//χρήση συνάρτησης ταξινόμησης
    printf("AFTER SORT RESULTS WILL SAVE TO FILE\n");
    save_to_file(employees,5);//Κλήση συνάρτησης για αποθήκευση σε αρχείο.
    free(employees);//Απελευθέρωση μνήμης που δέσμευσε ο δείκτης employees.
}
```

## ΠΑΡΑΔΕΙΓΜΑ\_2

- Να κατασκευαστεί μία δομή η οποία θα αναπαριστά προϊόντα. Η Δομή θα έχει σαν δεδομένα το αναγνωριστικό του προϊόντος(id), την κατηγορία(category) του προϊόντος και την τιμή του(price).
- Να κατασκευαστεί συνάρτηση η οποία θα εισάγει από ένα αρχείο τα δεδομένα για το προϊόν.
- Να κατασκευαστεί συνάρτηση η οποία θα δέχεται σαν όρισμα ένα προϊόν και θα κάνει αναζήτηση στον πίνακα για το προϊόν και θα εμφανίζει τα αντίστοιχα αποτελέσματα.
- Να κατασκευαστεί συνάρτηση η οποία θα δέχεται σαν όρισμα το όνομα της κατηγορίας ενός προϊόντος και θα εμφανίζει τα προϊόντα που υπάρχουν σε αυτήν την κατηγορία.
- Να κατασκευαστεί συνάρτηση η οποία θα εκτυπώνει όλα τα προϊόντα από το αρχείο.

## ΠΑΡΑΔΕΙΓΜΑ\_2

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
typedef struct
{
    int id;
    char category[100];
    double price;
}product;
char filename[100];
```



## ΠΑΡΑΔΕΙΓΜΑ\_2[Συνάρτηση εύρεσης μεγέθους αρχείου].

```
int getsizefromfile()
{
    FILE *fp;
    fp=fopen(filename,"r");
    if(fp==NULL)
    {
        return 0;
    }
    int cnt=0;
    char line[100];
    while(fgets(line,100,fp)!=NULL)
    {
        cnt++;
    }
    fclose(fp);
    return cnt;
}
```

## ΠΑΡΑΔΕΙΓΜΑ\_2[ΦΟΡΤΩΣΗ ΣΕ ΠΙΝΑΚΑ ΑΠΟ ΤΟ ΑΡΧΕΙΟ]

```
void loaddata(product *products)
{
    FILE *fp;
    fp=fopen(filename,"r");
    char line[100];
    int i=0;
    //strtok-->Διαχωρισμός αρχείου σε tokens[Διαφορετικές λέξεις ανάλογα με το σύμβολο διαχωρισμού].
    //επιστρέφει δείκτη σε χαρακτήρα αν εκτελεστεί επιτυχώς αλλιώς επιστρέφει NULL.
    while(fgets(line,100,fp)!=NULL)
    {
        char *temptok=strtok(line," -");
        products[i].id=atoi(temptok);
        temptok=strtok(NULL," -");
        strcpy(products[i].category,temptok);
        temptok=strtok(NULL," -");
        products[i].price=atof(temptok);
        i++;
    }
    fclose(fp);
}
```

## ΠΑΡΑΔΕΙΓΜΑ\_2[ΕΥΡΕΣΗ ΜΕ ΒΑΣΗ ΤΗΝ ΚΑΤΗΓΟΡΙΑ].

```
void search_by_category(product *products,int size,char *cat)
{
    int cnt=0;
    for(int i=0;i<size;i++)
    {
        if(strcmp(products[i].category,cat)==0)
        {
            printf("Id:%d \t Category:%s \t Price:%.2lf\n",products[i].id,products[i].category,products[i].price);
            cnt++;
        }
    }
    if(cnt==0)
    {
        printf("No such category found!!!\n");
    }
}

void display_products(product *products,int size)
{
    for(int i=0;i<size;i++)
    {
        printf("Id:%d \t Category:%s \t Price:%.2lf\n",products[i].id,products[i].category,products[i].price);
    }
}
```

## ΠΑΡΑΔΕΙΓΜΑ\_2[ΚΥΡΙΑ ΣΥΝΑΡΤΗΣΗ]

```
int main()
{
    strcpy(filename, "products.txt");
    int size=getsizefromfile();
    if(size==0)
    {
        printf("File doesn't exist!!!\n");
        exit(1);
    }
    product products[size];
    char categoryname[100];
    int flag=0;
    int choice;
    loaddata(products);
    int id;
```



## ΠΑΡΑΔΕΙΓΜΑ\_2[ΚΑΤΑΣΚΕΥΗ ΚΥΡΙΟΥ ΜΕΝΟΥ]

```
while(1)
{
    printf("MAIN MENU::\n");
    printf("1-Search product id\n");
    printf("2-Search proucts by category\n");
    printf("3-Display products\n");
    printf("4-Exit\n");
    printf("Give choice:");
    scanf("%d",&choice);
```

# ΠΑΡΑΔΕΙΓΜΑ\_2[ΕΠΙΛΟΓΕΣ]

```
switch(choice)
{
    case 1:
        printf("Give product id:");
        scanf("%d",&id);
        search(id,products,size);
        break;
    case 2:
        printf("Give category:");
        gets(categoryname);
        search_by_category(products,size,categoryname);
        break;
    case 3:
        display_products(products,size);
    case 4:
        flag=1;
        break;
    default:
        printf("invalid choice\n");
        break;
}
if(flag==1) break;
}
```

# ΑΣΚΗΣΕΙΣ

1. Για τους υπαλλήλους μιας εταιρείας θέλουμε να καταγράψουμε τα εξής στοιχεία: επίθετο, όνομα, ηλικία, φύλο, κιν. τηλέφωνο και μισθούς των τελευταίων 3 ετών.

- Να προτείνετε μία δομή (struct) C με το όνομα `employee` που να επιτρέπει την ομαδοποίηση αυτών των στοιχείων. Να γράψετε ένα πρόγραμμα C για την επεξεργασία των στοιχείων 10 υπαλλήλων.
- Εισαγωγή από το χρήστη των πιο πάνω στοιχείων για τον κάθε υπάλληλο.
- Για τον κάθε υπάλληλο υπολογισμός του μέσου όρου των μισθών των 3 τελευταίων ετών.
- Για το τελευταίο έτος, ο μέσος όρος των μισθών των υπαλλήλων.
- Τα στοιχεία του υπαλλήλου με τον μεγαλύτερο μισθό στο τελευταίο έτος.
- Εκτύπωση των δεδομένων σε αρχείο.

# ΑΣΚΗΣΕΙΣ

2. Για ένα σύνολο αυτοκινήτων υπάρχουν τα εξής στοιχεία: ☐ ΑΡΙΘΜΟΣ (μόνο το αριθμητικό μέρος): π.χ. 4352 , ΧΡΩΜΑ: π.χ. RED ,ΚΑΤΑΣΚΕΥΑΣΤΗΣ: π.χ. ALFA-ROMEO ,ΗΜΕΡΟΜΗΝΙΑ ΚΑΤΑΣΚΕΥΗΣ: π.χ. 2005 .Τα στοιχεία των αυτοκινήτων αποθηκεύονται στο αρχείο cars.txt.

Σε κάθε γραμμή του αρχείου υπάρχει μία εγγραφή με τα στοιχεία ενός αυτοκινήτου ως εξής: ,ΑΡΙΘΜΟΣ ΧΡΩΜΑ ΚΑΤΑΣΚΕΥΑΣΤΗΣ ΗΜΕΡΟΜΗΝΙΑ

- Να γραφεί πρόγραμμα C που αφού διαβάσει τα στοιχεία των αυτοκινήτων θα υπολογίζει:
- Τι ποσοστό των αυτοκινήτων κατασκευάστηκαν το 2006 .
- Τη διαφορά σε σχέση με το 2005 και το 2004.