université de science et technologie houari boumediene

TP 1: Structures (Records) in C

beldjouziasmaa@gmail.com

BELDJOUZI ASMAA
ingénieure en informatique

Introduction

ADRESSE EMAIL
beldjouziasmaa@gmail.com

An array consists of elements of the same type. However, it is often necessary to group elements of different types into a single entity to simplify manipulation. In the C language, we use structures (records) for this purpose. A structure is a finite sequence of elements that can have different types.
Each element of a structure, called a member or field, is designated by an identifier.

suivant →

BELDJOUZI ASMAA
ingénieure en informatique

Declaring a Structure

ADRESSE EMAIL
beldjouziasmaa@gmail.com

General Syntax:

```
struct idf {
    type-1 member-1;
    type-2 member-2;
    ...
    type-n member-n;
}
```

Remarks:

- This declaration defines a new type but does not allocate memory.
- All field names must be distinct within a structure.
- Two structures can have fields with the same name.

suivant →

BELDJOUZI ASMAA
ingénieure en informatique

Declaring a Structure

ADRESSE EMAIL
beldjouziasmaa@gmail.com

Declaring a Structure Type Variable:

```
struct idf object;
```

Or directly within the definition:

```
struct idf {
    type-1 member-1;
    type-2 member-2;
    ...
    type-n member-n;
} object;
```

suivant →

BELDJOUZI ASMAA
ingénieure en informatique

Accessing Structure Members

ADRESSE EMAIL
beldjouziasmaa@gmail.com

Structure members are accessed using the . operator.

```
object.member_i;
```

Example: Computing the modulus of a complex number.

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

struct complex {
    double real;
    double imaginary;
};

int main() {
    struct complex z = {3.0, 4.0};
    double modulus;
    modulus = sqrt(z.real * z.real + z.imaginary * z.imaginary);
    printf("Modulus of (%f + i %f) = %f \n", z.real, z.imaginary, modulus);
    return 0;
}
```

suivant →

BELDJOUZI ASMAA
ingénieure en informatique

Initializing a Structure

ADRESSE EMAIL
beldjouziasmaa@gmail.com

Initialization rules are similar to those of arrays.

```
struct complex z = {1.4, 2.5};
```

suivant →

BELDJOUZI ASMAA
ingénieure en informatique

ADRESSE EMAIL
beldjouziasmaa@gmail.com

Operations on Structures

Direct Assignment

Assignment is possible between two structures of the same type.

```
struct complex z1, z2;
z2 = z1;  // Copies fields from z1 to z2
```

Comparison

Structures cannot be compared directly. Each field must be compared individually.

suivant →

BELDJOUZI ASMAA
ingénieure en informatique

Defining Types with typedef

ADRESSE EMAIL
beldjouziasmaa@gmail.com

To simplify structure declarations, typedef can be used.

```
typedef struct {
    double real;
    double imaginary;
} complex;

int main() {
    complex z;
}
```

suivant →

BELDJOUZI ASMAA
ingénieure en informatique

Using Structures

ADRESSE EMAIL
beldjouziasmaa@gmail.com

Example structure for storing personal information:

```
typedef struct {
    char name[10];
    char firstname[10];
    int age;
    float grade;
} form;

form f1, f2;
strcpy(f1.name, "Badi");
strcpy(f1.firstname, "Ali");
f1.age = 20;
f1.grade = 11.5;
f2 = f1; // Copying structure
```

suivant →

BELDJOUZI ASMAA
ingénieure en informatique

ADRESSE EMAIL
beldjouziasmaa@gmail.com

Array of Structures

Structures can be used in arrays:

```
typedef struct {
    int Day, Month, Year;
} Date;

typedef struct {
    char Name[20], Address[30];
    Date Birth;
} person;

person T[20];
T[0].Birth.Day = 4; // Assigning a field
```

suivant →

BELDJOUZI ASMAA
ingénieure en informatique

Structures and Functions

ADRESSE EMAIL
beldjouziasmaa@gmail.com

Passing by Value

A structure can be passed to a function by value

```
void displayStudent(student std) {
    printf("Id: %d\n", std.id);
    printf("Name: %s\n", std.name);
    printf("Average: %f\n", std.average);
}
```

Passing by Address

Using pointers is more efficient to avoid unnecessary copies.

```
void readStudent(student *std) {
    printf("Enter Id\n");
    scanf("%d", &std->id);
    printf("Enter Name\n");
    scanf("%s", std->name);
    printf("Enter Average\n");
    scanf("%f", &std->average);
}
```

suivant →

BELDJOUZI ASMAA
ingénieure en informatique

Complete Example

ADRESSE EMAIL
beldjouziasmaa@gmail.com

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

#define NBR_STUDY 3

typedef struct {
    int id;
    char name[11];
    float average;
} student;

void displayStudent(student);
void readStudent(student *);

int main() {
    int i;
    student TabStudents[NBR_STUDY];

    for (i = 0; i < NBR_STUDY; i++) {
        readStudent(&TabStudents[i]);
    }
    printf("Displaying Students:\n");
    for (i = 0; i < NBR_STUDY; i++) {
        displayStudent(TabStudents[i]);
    }
    return 0;
}

void readStudent(student *std) {
    printf("Enter Id: ");
    scanf("%d", &std->id);
    printf("Enter Name: ");
    scanf("%s", std->name);
    printf("Enter Average: ");
    scanf("%f", &std->average);
}

void displayStudent(student std) {
    printf("Id: %d\n", std.id);
    printf("Name: %s\n", std.name);
    printf("Average: %f\n", std.average);
}
```

suivant →

BELDJOUZI ASMAA
ingénieure en informatique

Exercise 1

ADRESSE EMAIL
beldjouziasmaa@gmail.com

a- Define a TIME type which contains the hour, minute, second fields.
b- Write a TRANSFORM function which transforms a time T of type TIME into an integer S which expresses this time in seconds.
Example : for T=2 hours 10 minutes 37 seconds, S=7837 seconds.
c- Write a DECOMPOS procedure which decomposes a time S expressed in seconds into a time T of type TIME.
d- Given two times T1 and T2 of type TIME, write an algorithm which calculates the time T sum of times T1 and T2 (T, T1 and T2 are of type TIME). (use the TRANSFORM and DECOMPOS actions).

suivant →

BELDJOUZI ASMAA
ingénieure en informatique

Exercise 2

ADRESSE EMAIL
beldjouziasmaa@gmail.com

Consider a record E defined by two pieces of information:
- T an array of integers that can contain a maximum of 100 elements;
- N the number of elements of the array T.

Given a character string M , write a parameterized action which returns a record of type E containing all the positions of the string ' ab ' in the string M.

Using the Find parameterized action, write an algorithm to remove all occurrences of the string "ab".

Example : M = 'faabaababbaabrs'
Result :

3	6	8	12
	4		

suivant →