

## Lab 10 - Exercise – All sections

### structures

### CS 262 – Spring 2022

The main purpose of this activity is to practice structures in C and how to create new types using `typedef`.

Before creating structure variables, we must define the structure with the keyword **struct** and enclose the members of the structure in braces.

Syntax	Example
<pre>struct structureName{     dataType member1;     dataType member2;     ... };</pre>	<pre>struct coordinate{     int x;     int y; };</pre>

To declare structure variables, use **struct** and `structureName`. For instance, to declare the variables: `target1` and `target2` as **struct** `coordinate`, we need to write:

```
struct coordinate target1, target2;
```

Other alternative is to use the keyword **typedef** to create an alias name for the data types. This alternative is commonly used with structures to simplify the variable declaration syntax.

Syntax	Example
<pre>typedef struct {     dataType member1;     dataType member2;     ... };</pre>	<pre>typedef struct{     int x;     int y; }coordinate;</pre>

When using `typedef` to define a new type of structure, we only need to specify the new type and the variable name.

```
coordinate target1, target2;
```

To access a member of a structure, specify the variable name, the dot operator (`.`), and the member. Here is an example of how we could assign 10 to the `x` member in `target1`.

```
target1.x = 10
```

To create an array of structures, specify the type, name of the array and the array length enclosed in brackets. Example how to create an array of 3 elements of type `coordinate`.

```
coordinate targets[3];
```

To access a member of a structure in an array of structures, indicate name of the array, enclosed in brackets the index of element of the array you want to access, after brackets use the dot operator (`.`) and the member's name. Here an example how we could to assign 50 to `x` in the first element of the `targets` array.

```
coordinate targets[0].x = 50;
```

## Description of the program

Code a C program that gets the information from "N" drivers to store them in an array of structures. Then print the stored information starting **from the last to the first driver**.

- Ask the user to enter an integer (N) to create an array of structures using dynamic memory, where ( $2 \leq N \leq 5$ ).
- Use a for-loop to enter the data for each driver.
- Print the stored information starting from the last driver

### Notes:

- Use **typedef** to define the new type "**driver**" which is a structure with two members: `id(int)` and `name (array of 35 char)`.
- Use `fgets()` and `sscanf()` to get the user input.
- To simplify the activity, it is not necessary to validate the information to avoid duplicates, i.e., the `id` of the drivers could be the same in different drivers, etc.

### Hints

- Use `%[^\n]` instead of `%s` when using `sscanf()` to get blank spaces in `name`

### Example:

```
How many drivers you want input? 2
```

```
Driver 1, id: 234
```

```
Driver 1, name: Ironman
```

```
Driver 2, id: 123
```

```
Driver 2, name: Captain America
```

```
Driver's Licenses:
```

```
123, Captain America
```

```
234, Ironman
```

```
That's all! BYE :)
```