Hands-on Exercises

(Strings, Dynamic Allocation and Structures)

IT 1201

The problems are found inside my folder in Drive T: (teacherfiles)

Instructions:

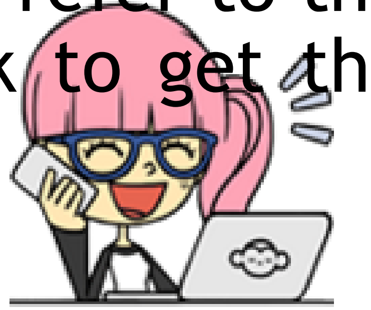
1. In My Documents, create a folder with your last name and the date today as your folder name. (e.g. Polinar_Dec3)
2. Solve each problem one by one and save your outputs on the folder you created.
3. Once you are done with a problem and are SURE that your answer is correct, call the attention of you teacher to check your output.

Save as: sameName.c

Given the structure definition of a complete name:

```
typedef struct Name {
    char fN[24]; /* First name */
    char lN[16]; /* Last name */
    char mI; /* Middle Initial */
}Nametype;
typedef enum {FALSE, TRUE} boolean; /* information on enum
is found on the next slide */
```

Write a program that will invoke the function `isSameName()`. The function will return TRUE if the 2 given complete names refer to the same person, otherwise return FALSE. It is main's task to get the necessary inputs and display if the names are the same or not..



Enumerated Types

- Allow us to create our own symbolic names for a list of related ideas
- The key word for an enumerated type is **`enum`**
- Enumerated types are NOT STRINGS!
  - Even though enumerated type values look like strings, they are new key words that we define for our program.
- They are treated *like* integers by the compiler.
  - Underneath they have numbers 0,1,2,... etc.
  - Example:

```
enum rank {First, Second, Third};
/* First = 0, Second = 1, Third = 2 */
```

Source: http://www.cs.utah.edu/~germain/PPS/Topics/C_Language/enumerated_types.html

Save as: userName.c

Create a C program that would include the following:

- ❖A structure named `studDetails` that would hold the student's first name, last name, and his programming subject's code.
- ❖A function named `getAccount()` which will return to the calling function, the user account; given the structure variable containing the student's details. The user account should follow the format below:

```
<lastName><first2LettersOfFirstName>_<subjCode>
```

For example:

INPUT			OUTPUT
firstName	lastName	subjCode	polinarst_IT1201
stephanie	polinar	IT1201	

Save as: myBday.c

```
typedef struct {
     int month, day, year;
}birthday;
typedef struct {
     char fN[24];
     char lN[16];
     char mI;
     birthday bday;
     int age;
}student;
```

Given the structure definition found in the box on the right, create a C program that would call the function `getBday()` and `dispBday()`.

- `getBday()` - will accept a student record and let the user input a last name. If the input is the same as the last name in the student record, then the function will return the student's birthday. Otherwise, the function should display "`You have the wrong student record!`" and terminate the program.
- `dispBday()` - will simply display the given birthday in the following format: month/day/year.

The `main()` function is partially done on the next slide.

Prepared by: adcantara-sbpolinar

Save as: myBday.c

```
int main(void)
{
    student x = {"Setsuna", "Seiei", 'F', 4, 7, 2291, 16, 'M'};
    birthday birth;

    ____________________; //call to getBday()

    printf("\n%s's birthday is ", x.fN);

    ____________________; //call to dispBday()

    getch();
    return 0;
}
```