Practical worksheet N°3. Conditional structures

1) Objectives

This practical work aims to learn, through a set of activities and exercises, the different forms of conditional structures and their use.

At the end of this practical work, you should be able to solve any problem containing multiple situations and use all the conditional instructions studied correctly.

2) Tests: simples, compound, and nested

Consider the following program:

```
#include <stdio.h>
int main(){
    int n;
    printf("Enter an integer number: ");
    scanf("%d",&n);
    if(n%2==0)
      printf("The number %d is divisible by 2",n);
    if(n%2!=0)
      printf("The number %d is not divisible by 2",n);
    return 0;
}
```

a) Create a new project and type the program above.
b) Compile and run it. What does this program do?
c) Modify the program to perform the same task but with a single comparison.
d) Make the necessary adjustments to the program to test whether the entered number is divisible by 2 and 3, only by 2, only by 3, or by neither of them.

3) Multiple-choice structure

We want to write a program that displays, for a selected season, the list of months it contains. The program starts by displaying, in a menu, all the seasons of the year. The user is then prompted to select one of the seasons. Finally, the program displays the months of that season.

Recall that there are four seasons (spring, summer, autumn, and winter), each containing three months, and that March is the first month of spring.

The menu to be displayed is in the following format:

```
List of seasons:
    1: Spring
    2: Summer
    3: Autumn
    4: Winter
Enter your choice:
```

If the user, for example, enters 3, the program displays:

```
The months are: September, October, November
```

The following code, once completed, is supposed to perform the requested task:

```
#include <stdio.h>
int main(){
    int s;
    printf("List of seasons:\n");
    printf("\t1: Spring\n");
    ..............................
    ..............................
    switch(s){
        case 1: ................
                break;
        ........................
        default: ...............
    }
    return 0;
}
```

- Create a new project and type the code provided above.
- Complete the program to make it perform the desired task, then compile and run it.

4) Branching statement

Using the **`goto`** branching instruction, write a C program that allows to enter a sequence of positive integer numbers (the sequence ends when a negative or zero value is entered) and calculates and displays the sum of the entered numbers **(The algorithm was covered in lecture)**.

5) Application exercises

1. Write a program for entering 3 integer numbers and displaying them in ascending order. **(The algorithm was covered in TW)**.
2. Establish a program that reads the coordinates (x, y) of a point and determines if this point is inside a rectangle defined by the coordinates of its top-left point ($x1$, $y1$) and its bottom-right point ($x2$, $y2$). **(The algorithm was covered in lecture)**.
3. Write the program to solve a second-degree equation of the form:

$$a\,x^2 + b\,x + c = 0.$$

4. Write a program that reads a character from the keyboard and determines whether it is a letter, a digit, or a symbol.
5. A store offers discounts to its customers based on the purchase amount under the following conditions:

   - If the purchase amount is less than 2000 DA, there is no discount.
   - If the purchase amount is between 2000 DA and 5000 DA, the amount above 2000 DA is subject to a 10% discount.
   - If the purchase amount is greater than 5000 DA, in addition to the previous discount, the customer receives an additional 20% discount for the amount above 5000 DA.

   Write a program that reads the purchase amount and calculates and displays the net-to-pay, where:

   Net-to-pay = purchase amount - discount amount

   (The algorithm was covered in TW).
6. Write a program that reads a time in the format (hour:minutes:seconds) and displays the time one second later.

   For example, if the user enters 21:32:8, the program should display: “One second from now, it will be 21:32:9.
7. Write a program that asks for an amount of money between 1 and 100 DA and then displays the minimum number of coins (50, 10, 5, and 1) required to make up that amount.
8. Write a program that allows to enter a day of the week (a number between 1 and 7) and display the corresponding day's name. For example, Saturday corresponds to the number 1.
9. Write a program that, from a menu, allows you to perform addition, subtraction, multiplication, or division of two numbers based on the user's choice. The two numbers and the operation to be performed should be entered by the user. **(The algorithm was covered in TW)**.

10. Write a program that repeatedly prompts for a student's mark (between 0 and 20) until a valid response is given. Use branching statements to achieve this. **(The algorithm was covered in TW)**.

6) Additional exercises

The exercises in this section are additional self-study exercises.

11. We say that two numbers are related if they are opposites or inverses. Two numbers are opposites if their sum equals 0, and inverses if their product equals 1.

    Write a program that reads two real numbers and determines if they are related or not. **(The algorithm was covered in TW)**.

12. Write a program that asks the user for a child's age and determine its sports category. The categories are :

    - U8 (Under-8) : from 6 to 7 years old,
    - U10 (Under-10) : from 8 to 9 years old,
    - U12 (Under-12) : from 10 to 11 years old,
    - U18 (Under-18) : 12 years old and older.

13. Write a program that allows you to enter a date in the format (day/month/year) from the keyboard and checks whether this date is valid or not.

    Remember that the month of February has 28 days, except in leap years, when it has 29. A year is a leap year if it is divisible by four but not by 100, or if it is divisible by 400. For example, 1980 and 1996 are leap years because they are divisible by 4 but not by 100. The year 2000 is a leap year because it is divisible by 400. However, the years 2100 and 3000 are not leap years because they are divisible by 100.

14. Write a program that allows to read the values of two boolean variables, 'a' and 'b', and check the two laws of Morgan:

$$\left(\overline{a+b}\right)=\overline{a}\cdot\overline{b}$$

$$\left(\overline{a\cdot b}\right)=\overline{a}+\overline{b}$$

15. Three switches, 'a,' 'b,' and 'c,' control the lighting of two lamps, 'R' and 'S,' following the following conditions:

    - As soon as one or more switches are activated, lamp 'R' must turn on.
    - Lamp 'S' should only turn on if at least two switches are activated.

    Write a program that allows input of the states of the 3 switches (1 for on and 0 for off) and then determines if lamps 'R' and 'S' are on or off.