Week 2

Exercise 1: Selection Structure, Menu-Based Program

Write a program that prompts the user for two integers and performs a calculation operation on them based on a simple menu. The options are to **add** the numbers together and **multiply** them. Perform the above operations using the `if-else` structure based on the selection. Print a description of the operation performed and the result as shown in the examples below. Your program should handle *unsuitable* menu selections as well.

Add a newline to the end of each line.

Example use case:

```
Enter two integers:
2 4
1: Add numbers
2: Multiply numbers
1
Numbers were added together. Result = 6.
```

Example use case:

```
Enter two integers:
3 5
1: Add numbers
2: Multiply numbers
2
Numbers were multiplied. Result = 15.
```

Example use case:

```
Enter two integers:
4 5
1: Add numbers
2: Multiply numbers
3
Unknown selection.
```

Exercise 2: For-loop

Write a program that prompts the user to enter a number between 10 and 200. Then it sums the numbers from zero to the requested number and prints the result according to the example below. Be sure to check that the given number is within the range 10-200. Alert the user if the number does not belong to the range.

Add a newline to the end of each line

Example use case:

```
Enter a number between 10 and 200:
65
The sum of the numbers 0 to 65 is 2145.
```

Example use case:

```
Enter a number between 10 and 200:
265
The number you entered is not within the given range.
```

Exercise 3: While-loop

Write a program that prompts the user to enter an integer between 1 and 10. Then it prints all the numbers between 1 and the given number according to the example below. The program should check that the given number belongs to the correct range and use **while-**loop in printing.

Example use case:

```
Enter an integer from 1 to 10:
6
Round 1...
Round 2...
Round 3...
Round 4...
Round 5...
Round 6...
```

Example use case:

```
Enter an integer from 1-10:
25
The number you entered is not between 1-10.
```

Exercise 4: switch-case

Write a program that asks the user for a floating-point number. Then it displays a simple menu with two options of multiplicatoin and division (see examples below). Your program should handle *unsuitable* menu selections.

Use `switch-case` for selection. In addition, use *preprocessor* to set the **constant** value for `Pi` and use that constant in calculations. Use three decimal places for `Pi` and use floating-point numbers, that is, 3.141. For printing, leave the total width of the floating-point numbers undefined, that is, use "`%.3f`" formatting.

Add a newline to the end of each line.

Example use case:

```
Enter a floating-point number:
1.7
MENU
1: Multiply Pi by 1.700.
2: Divide Pi by 1.700.
1
Pi * 1.700 = 5.340.
```

Example use case:

```
Enter a floating-point number:
1.8
MENU
1: Multiply Pi by 1.800.
2: Divide Pi by 1.800.
2
Pi / 1.800 = 1.745.
```

Example use case:

```
Enter a floating-point number:
1.5
MENU
1: Multiply Pi by 1.500.
2: Divide Pi by 1.500.
6
Unknown selection.
```

Exercise 5: Simple text editor with error checking

In C, the processing of a string is happening on the allocated memory area. The end of the strings are marked by the ending character "`\0`"

Write an "editor" program in which strings are handled differently by reserving a character array and maintaining the length of the stored string in a variable. This means that there is no ending character in the string.

You should be able to add new characters to the end of the string and delete the last character. You should also be able to print the string. **The program should be running until the user stops it.**

Your program should have the following menu:

```
1: Add a new character
2: Delete the last character
3: Print the string
0: Stop
```

At the beginning of the program, reserve a character array of 200 characters. Program the above functions and use switch-case for controlling what to do.

Adding a character means storing the character at the next free location in the array and increasing the size by one.

The last character can be deleted just by decreasing the size by one.

The characters in are printed one-by-one using `for`-loop.

You should be able to detect and report the following error situations:

Incorrect menu entry. Print: `Unknown selection.`

If there are **no characters** in the string when it is printed of the a character should be deleted, print: `The string is empty.`

It is also possible that the space of the 200 characters is out. Then print: `The space is out.`

Note that `stdio.h` is the only required header file

Add a newline to the end of each line.

Example run:

```
MENU
1: Add a new character
2: Delete the last character
3: Print the string
0: Stop
Your selection:
5
```

```
Unknown selection.
MENU
1: Add a new character
2: Delete the last character
3: Print the string
0: Stop
Your selection:
1
Enter a character:
Y
MENU
1: Add a new character
2: Delete the last character
3: Print the string
0: Stop
Your selection:
1
Enter a character:
e
MENU
1: Add a new character
2: Delete the last character
3: Print the string
0: Stop
Your selection:
1
Enter a character:
s
MENU
1: Add a new character
2: Delete the last character
3: Print the string
0: Stop
Your selection:
3
Yes
MENU
1: Add a new character
2: Delete the last character
3: Print the string
0: Stop
Your selection:
2
The character s has been removed.
MENU
1: Add a new character
2: Delete the last character
3: Print the string
0: Stop
Your selection:
3
Ye
MENU
1: Add a new character
2: Delete the last character
3: Print the string
```

```
0: Stop
Your selection:
2
The character e has been removed.
MENU
1: Add a new character
2: Delete the last character
3: Print the string
0: Stop
Your selection:
2
The character Y has been removed.
MENU
1: Add a new character
2: Delete the last character
3: Print the string
0: Stop
Your selection:
3
The string is empty.
MENU
1: Add a new character
2: Delete the last character
3: Print the string
0: Stop
Your selection:
2
The string is empty.
MENU
1: Add a new character
2: Delete the last character
3: Print the string
0: Stop
Your selection:
0
End of the program.
```