

## Exercises about control structures

- Solve them in Visual Studio or use <https://dotnetfiddle.net/>.

### *Exercise 06.01*

- Loop asking a number, no decimals.
- If it is not a number, ask again.
- So, you have a number.
- Show some text on the console.
- If number is divisible by 5.
  - Show the text “Divisible by 5”.
- If not.
  - Show the text “Not divisible by 5”.



*You can use Modulo (%) to check if it is divisible.*

- *15 % 5 has a result 0, so it is divisible.*
- *10 % 6 has a result 4, so it is not divisible.*

### *Exercise 06.02*

- Same exercise as 06.01.
- Use the ? operator.

Notes

---

---

---

---

---

---

---

---

COPY PASTE)

*Exercise 06.03*

- Similar to 06.01, but some extra tasks.
- Loop asking a number, no decimals.
  - If it is not a number, ask again, after showing an error message.
- So, at the end, you have a number.
- Loop asking a second number, no decimals.
  - If it is not a number, ask again, after showing an error message.
- So, at the end, you have a second number.
- Show some text on the console.
- If number is divisible by second number.
  - Show the text “Divisible by x”. (x is the second number)
- If not.
  - Show the text “Not divisible by x”. (x is the second number)



*You can use Module (%) to check if it is divisible.*

- *15 % 5 has a result 0, so it is divisible.*
- *10 % 6 has a result 4, so it is not divisible.*

*Exercise 06.04*

- Ask 5 numbers.
- Multiply them all.
- Show on the console if the result is negative or positive.

Notes

---

---

---

---

---

---

---

---

COPY PASTE)

### *Exercise 06.05*

- Ask for an age.
- Only positive numbers lower than 100 are allowed.
- If not correct.
  - Ask again after showing an error message.
- Depending on the age you show a different message on the console.
  - Younger than 13 → You are a child.
  - Younger than 18 → You can't drive a car yet.
  - Younger than 66 → You are an adult.
  - Other situations → You are an elder.

### *Exercise 06.06: C# if*

- [https://www.w3schools.com/cs/exercise.asp?filename=exercise\\_conditions1](https://www.w3schools.com/cs/exercise.asp?filename=exercise_conditions1)
  - Exercise 1.
  - Exercise 2.
  - Exercise 3.
  - Exercise 4.
  - Exercise 5.

### *Exercise 06.07: C# switch*

- [https://www.w3schools.com/cs/exercise.asp?filename=exercise\\_switch1](https://www.w3schools.com/cs/exercise.asp?filename=exercise_switch1)
  - Exercise 1.
  - Exercise 2.

Notes

.....

.....

.....

.....

.....

.....

.....

COPY PASTE)

*Exercise 06.08: C# while*

- [https://www.w3schools.com/cs/exercise.asp?filename=exercise\\_while\\_loop1](https://www.w3schools.com/cs/exercise.asp?filename=exercise_while_loop1)
  - Exercise 1.
  - Exercise 2.
  - Exercise 3.
  - Exercise 4.
  - Exercise 5.

*Exercise 06.09*

- Ask for country code.
  - Depending on given code (use switch)
    - "B" or "b", show "Belgium" and "Brussels".
    - "N" or "n", show "Netherlands" and "Amsterdam".
    - "F" or "f", show "France" and "Paris".
    - "G" or "g", show "Germany" and "Berlin".
    - All other cases, show "Unknown" and "Unknown".
- Call a method "NestedIfs". (You have to create this in the next steps)



Think "Karel the robot".

Look how the method "Main()" is set up.

- Create a new method "AskForAnAge".



```
static void AskForAnAge()
{
    Here you do your stuff
}
```

Notes

---

---

---

---

---

---

---

---

---

---

COPY PASTE)

- Ask for an age in that method.
  - Depending on given age (use if)
    - Lower than 18, show text “You are too young”).
    - Larger than 80, show text “You are too old”).
    - All other cases, show text “Welcome”).

### *Exercise 06.10*

- Ask for texts.
  - As long as the text is not “quit”.
  - Continue asking texts.
- The moment you have typed “quit”.
  - Show all the entered texts, every text on a new line in the console.

### *Exercise 06.11*

- Ask for 10 numbers.
- Show in the console the biggest number.
- Show in the console the smallest number.
- Show in the console the average.

### **Example**

- When the numbers are 10, 8, 6, 4, 2, 1, 3, 5, 7 and 9.
  - Largest: 10.
  - Smallest: 1.
  - Average: 5,5.

Notes

---

---

---

---

---

---

---

---

COPY PASTE)

*Exercise 06.12*

- Show in the console all numbers that are divisible by 3 and lower than 1000.
  - All on the same line (if possible) with a comma and a space between them.

**Example**

0, 3, 6, 9, 12, 15, 18, 21, ... 993, 996, 999



*Can you do it backwards?*

999, 996, 993, ..., 6, 3, 0

*Exercise 06.13*

- You start with an amount of 2500 euro.
- You invest this amount and you gain 7% every year.
- Show me the current amount for every year.

**How to calculate the correct amount?**

- You have your start capital. In this example 2500 euro.
- You multiply this with:
  - $(1 + \text{gain in percentage})$  to the power year.

So, for the second year you have:

$$2.862,25 = 2.500 * (1 + 0,07)^2$$

**Example**

The fourth year you have around 3.277 euro. This number is rounded.

*Notes*


---

---

---

---

---

---

---

---

---

---

COPY PASTE)

*Exercise 06.14*

- You ask for a text.
- Show every letter of the text in reversed order.



*Use a for loop.*

*This resembles on the exercise 0006-p Declaration.zip.*

*Exercise 06.15*

- You ask an amount of the type “decimal”. So, digits after the decimal symbol are allowed.
- This amount can’t be negative.
  - If it is, re-ask.
- If the input is not an amount.
  - Re-ask.
- You round that amount towards the second decimal.
  - 100,123 → 100,12.
  - 100,567 → 100,57.



*Use decimal.Round(variable, 2)*

- Now you have to calculate how you can pay this amount, always with the less paper and coins possible.
- Show on the console the following text, but only if it is needed.
  - xx times 500 (xx must be larger than 0 to be shown).
  - xx times 200.

Notes

---

---

---

---

---

---

---

---

COPY PASTE)

## Notes

---

---

---

---

---

---

---

---

COPY PASTE)

○ ...

- The possible amounts are  $500 - 200 - 100 - 50 - 20 - 10 - 5 - 2 - 1 - 0,5 - 0,2 - 0,1 - 0,05 - 0,02 - 0,01$ .

## Example

- When the amount is 645,54 this should be the result.

```
1 time 500
1 time 100
2 times 20
1 time 5
1 time 0,5
2 times 0,02
```

## Exercise 06.16



*Take your time for this exercise.*

*Try to estimate before you start in how long it will take you.*

*Make post-its (tasks) of the steps that you want to execute.*

*Do the exercise first manually to see what the asked routine is.*

*You will be forced to work in a structured way. If you don't, you will have a lot of troubles. Do also small steps and test every time the result you have at the moment.*

*Working code is not good enough, it must also be maintainable.*

- You ask a text that is a series of “0”s and “1”s.
  - E.g., 00110101
- As long you have other characters in the text, you re-ask the text.
- The moment you have a good starting point, you do this.
  - Loop thru the text and create another text.
    - When the character is “1”, you add “10”.
    - When the character is “0”, you add “01”.



- So, the example 00110101 results in during the looping.
  - 0 (first character) → 01.
  - 0 (second character) → 0101.
  - 1 (third character) → 010110.
  - 1 (fourth character) → 01011010.
  - And so on.
  - End result: 0101101001100110.
- With that end result, you do the same routine again.
  - Loop thru the text and create another text.
    - When the character is “1”, you add “10”.
    - When the character is “0”, you add “01”.
- You should now have a text of 32 symbols.
- With this result you do this.
  - Every second occurrence of a 0, you remove it.
  - Every second occurrence of a 1, you remove it.
- In other words:
  - The second, fourth, sixth, and so on ‘zero’ will be removed.
  - The second, fourth, sixth, and so on ‘one’ will be removed.
- Let’s assume that the result is 1101010000.
  - This will result in ~~1+010+0000~~.
  - The end result will be 10100.
- What do you notice if you do this for the given start example 00110101?



*The goal is first to write all the code in the Main method.*

*Later on, we will divide this code in different methods and routines.*

*The goal of the exercise is to compare the way of solutions. One with only one method, the other with several methods.*

Notes

.....

.....

.....

.....

.....

.....

.....

COPY PASTE)



*Does your routine work if you type nothing?*

*Does your routine work if you type 0?*

*Does your routine work if you type 1?*

*Does your routine work if you type 00?*

*Does your routine work if you type 01?*

*Does your routine work if you type 10?*

*Does your routine work if you type 11?*

*Does your routine work if you type abc?*

Notes

---

---

---

---

---

---

---

---

COPY PASTE)

*Exercise 06.17*

- You ask for a number, no decimals.
  - 12.548
- You count the sum of all numbers from 1 till the number that was entered.
- Try to measure how long this routine takes to find a solution.
  - Use `DateTime.Now` at the start and end of your routine.

**Example**

- When you enter 12.548, your result should be 78.732.426.
- When you enter 12.345.678 your result should be 76.207.888.812.681.
- When you enter 123.456.789 your result should be 7.620.789.436.823.655.
- When you enter 1.234.567.890 your result should be 762.078.938.126.809.995.



*How did you solve this?*

*Ask Vincent for the fastest way.*

Notes

---

---

---

---

---

---

---

---

COPY PASTE)