EXERCISE 1

- Enter a number (n1) on the console.
- Enter a number (n2) on the console.
- Compare the two number:
    - If n2 is greater than n1 print the sum of the two numbers.
    - Otherwise print *-1*.

Q1: What will be the **result** for these outputs?

Input	Output
> 3 > 8	8 (8 = 3 + 5)
> 5 > 5	-1
> 10 > 8	-1
> 0 > 5	5
> 1 > 3	4

Q2: **Analyze the symbols** you need to solve this problem.

Element	Do you need it?	For what?
Action 		
Decision 		
Repeat 		
Input / Output 		

Q3: From the following code, draw the corresponding flowchart.

```
n1 = int(input())
n2 = int(input())
sum = -1
if n2 > n1:
   sum = n1 + n2
print(sum)
```

Q4: Solve this poll of question on hackerrank:

Answer:

```
n1 = int(input())
n2 = int(input())
sum = -1
if n2 > n1:
    sum = n1 + n2
print(sum)
```

https://www.hackerrank.com/work/tests/1404754/questions

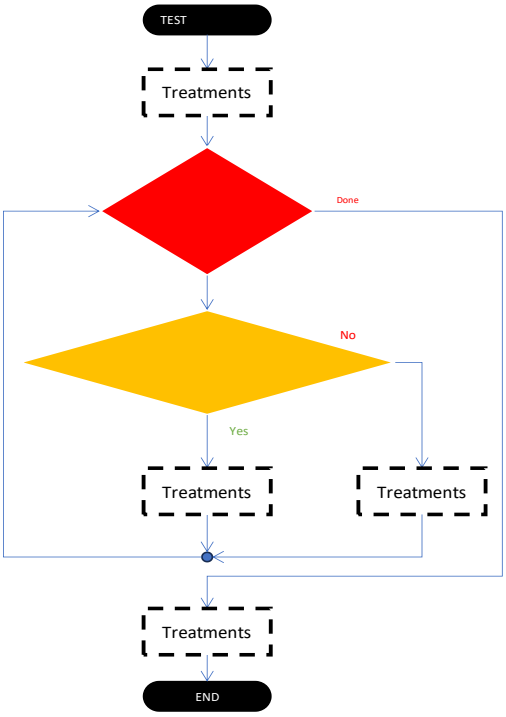
Exercise 2

- Check if string contains only “A”.

Q1: What will be the **result** for these outputs?

Input	Output
BBBB	False
AA	True
BBBBA	
AABC	
AAAAAAA	
AAAD	

Q2: Fill up the gap of the following flowchart.



Implement it in Python.

Answer:

```
n=input()
result=False
for i in range(len(n)):
    if n[i]=="A" or n[i]=="a":
      result=True
print(result)
```

Q3: Execute it and fill up the following result table.

Input	Output
BBBAB	True
AAAAAA	True
ABABBBA	True
AABCA	True

Exercise3

- Enter a number again and again until there is no “end” string.
- Print the even number in the console.

Q1: What will be the **result** for these outputs?

Input	Output
> 5 > 3 > 2 > 10 > end	2:10
> 5 > 4 > 8 > 16 > end	4:8:16
> 2 > 6 > 4 > 7 > end	2:6:4
> 5 > 3 > 7 > 1 > end	
> 1 > 2 > 3 > end	

Q2: Create a flowchart by using the condition (while) block.

Q3: Implement it in Python.

Answer:

```
text=""
result=""
while text !="end":
    text=input()
    if text!="end" and int(text)%2==0:
        result+=text+":"
print(result[:-1])
```

Q4: Is it possible to adapt your solution by using the repeat block? If so, do it.

Exercise4

- Input text in the console
- Print index of the LAST pattern “KK” (first K letter)
- If no “KK”, write -1

Q1: What will be the **result** for these outputs?

Input	Output
DDKDDDKE	-1
DDKKDDKKD	6
K	-1
AAKAKK	4
XXKKXXKX	2
KKAAK	0

Q2: What is wrong with this code?

```
text = input()
result = -1
for index in range(len(text)):
   letter = text[index]
   if letter == "K" and and text[index + 1] == "K":
      result = index
print(result)
```

Q3: To find the bug, test this code and check if the output is correct.

INPUT	OUTPUT
KK	0
KKK	1
AAKKAK	2

Q4: Write the correct code.

```
text = input()
result = -1
for index in range(len(text)-1):
```

```
    letter = text[index]
    if letter == "K" and text[index + 1] == "K":
        result = index
print(result)
```

Exercise 5

- Enter 2 numbers using a string, as follows: “number1; number2”.
- Number1 and number2 shall be composed of 2 digits (example “45”)
- Between number1 and number2, the character “;” shall be entered.
- If format is good
    - Display the sum of these 2 numbers.
    - Otherwise display: “WRONG FORMAT”

Q1: What will be the **result** for these outputs?

Input	Output
12;13	25
Ronan	WRONG FORMAT
12;1	WRONG FORMAT
10;20	30
12/14	WRONG FORMAT
15;16;18	49

Q2: Do you need extra variables?

Q3: What is the information we need to keep?

Q4: Solve this problem and try your code with the previous inputs.

```
text=input()
sum=0
if len (text)==5 and text[2]==";":
    sum+=int(text[:2])+int(text[3:])
    print(sum)
else:
    print("Wrong Format")
```