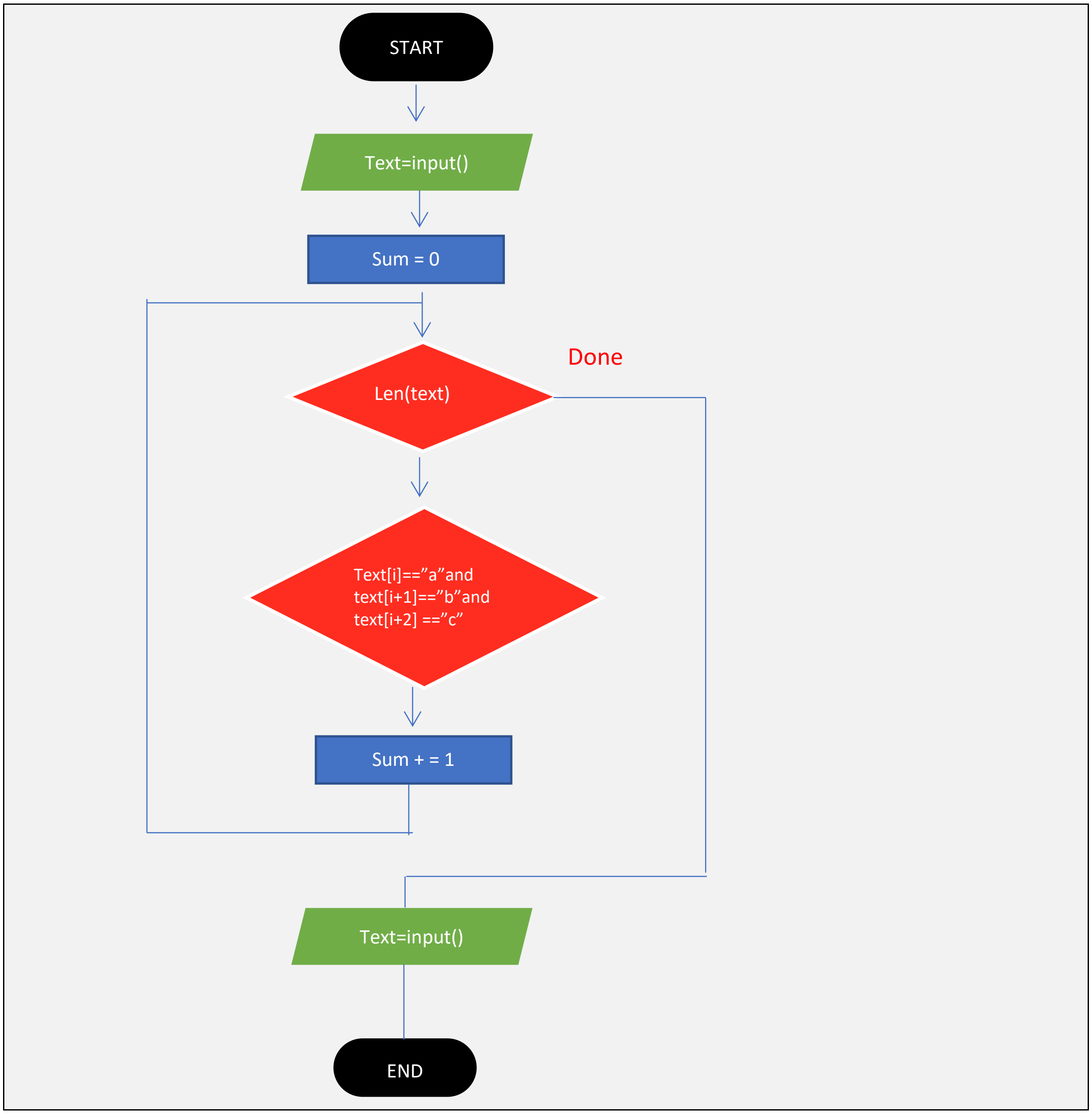
EXERCISE 1

- Input a text in the console.
- Count the number of times the "abc" pattern appears in a string.

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

Input	Output
abbcdef	0
abcfabc	2
adabcfbc	0

Q2: Choose the more relevant blocks and draw your structure flowchart.



Q3: Adapt code from a previous session to solve this problem.

```
text=input()
sum=0
for i in range(len(text)):
    if text[i]=="a"and text[i+1]=="b"and text[i+2]=="c":
        sum+=1
print(sum)
```

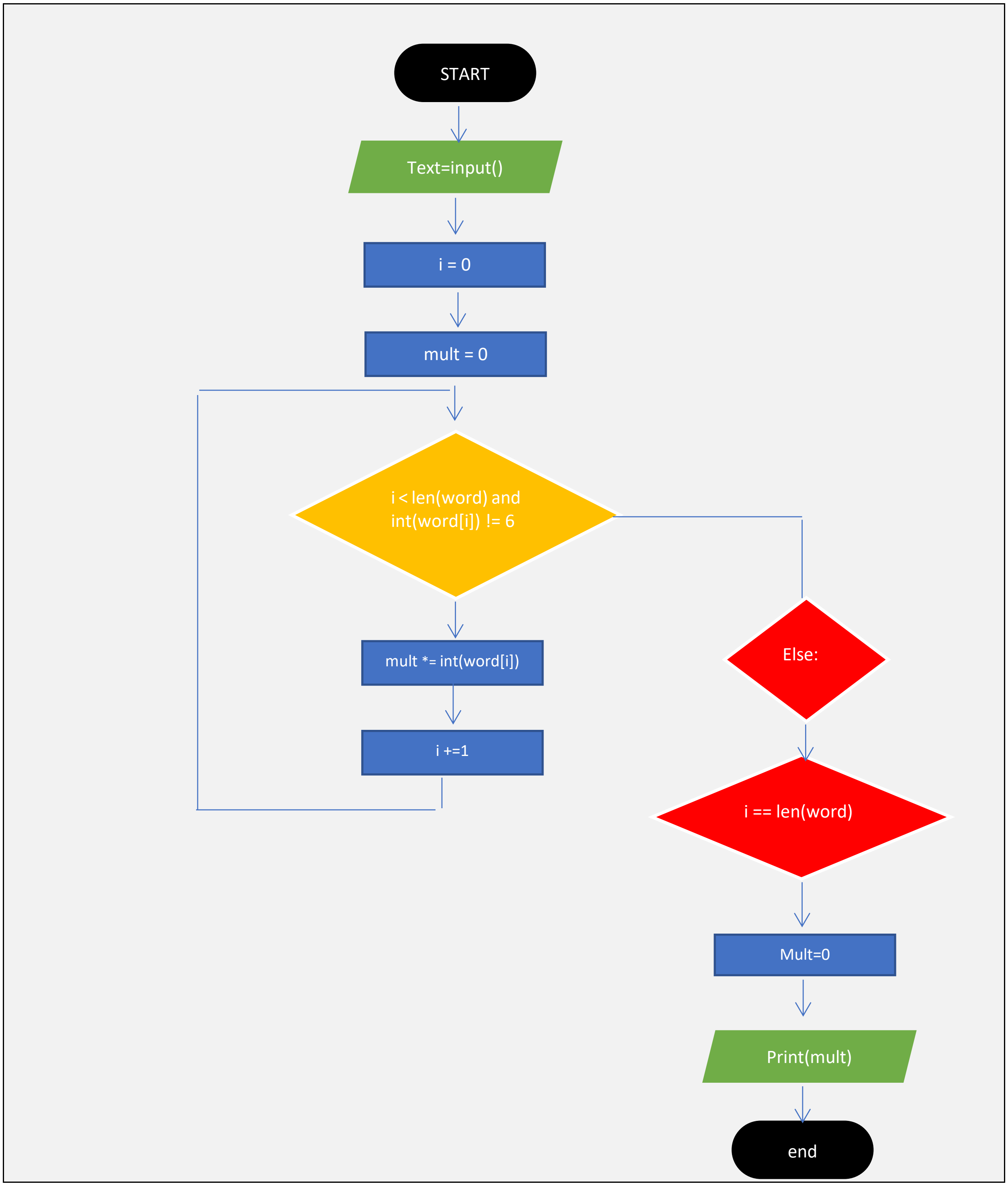
EXERCISE 2

- Input a text in the console, only number.
- Multiple all the number before the first apparition of 6 and 1 if 6 is in the first position.

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

Input	Output
123456789	120
5263	10
71256	70
45	0
625	1

Q2: Draw your structure flowchart.



Q3: Use the following code and adapt it so as not to use the repetitive (for) loop.

```
word = input()
i = 0
mult = 1
while i < len(word) and int(word[i]) != 6:
    mult *= int(word[i])
    i += 1
print(mult)
```

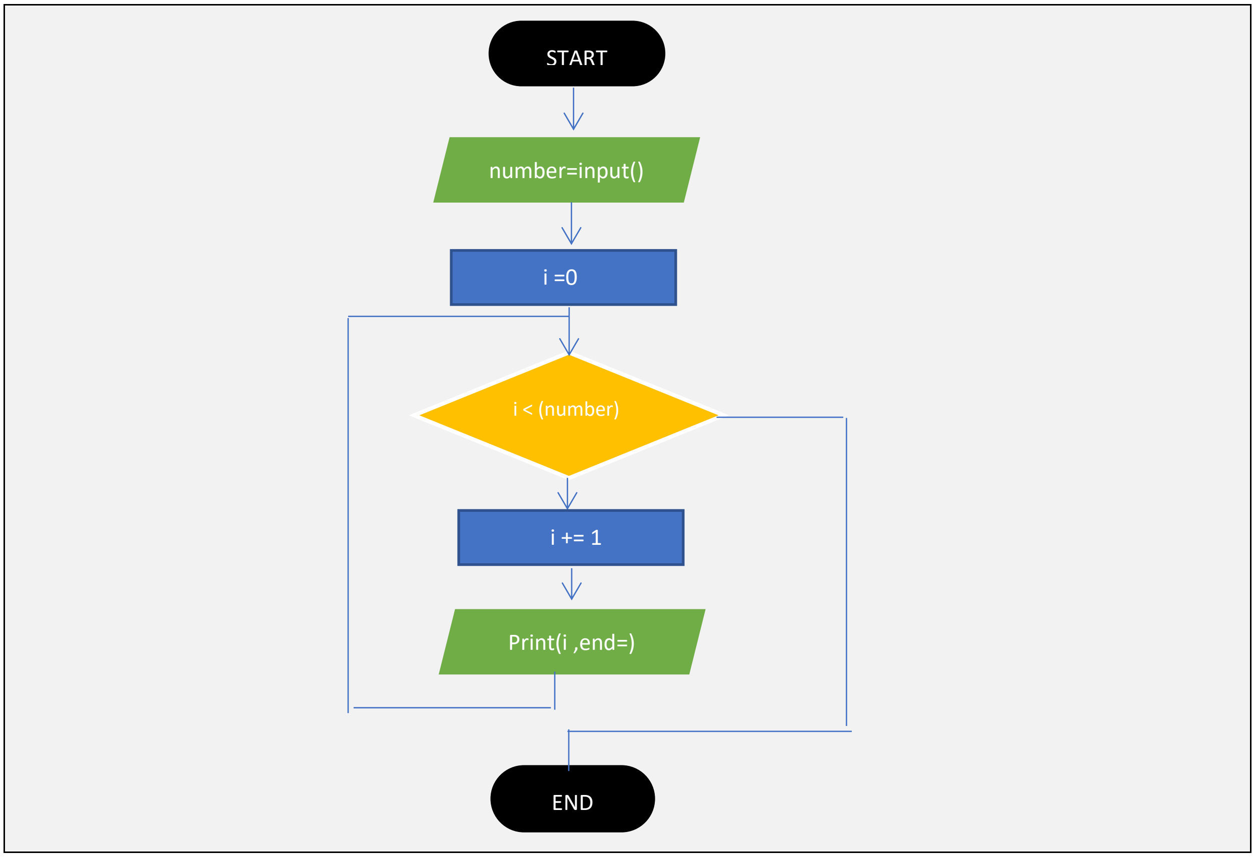
```
word = input()
i = 0
mult = 1
while i < len(word) and int(word[i]) != 6:
    mult *= int(word[i])
    i += 1
else:
    if i == len(word):
        mult = 0
print(mult)
```

EXERCISE 3

- Input a number.
- Display numbers from 1to that number

Input	Output
4	1 2 3 4
2	1 2
5	12345
3	123
0	0

Q1: Create a flowchart to solve this problem. Which loop block will you choose and why?



Q2: Review the code and find **the error** and explain them.

```
number = int(input())
result = ""
for index in range(number):
  result = result + " " + str(index + 1)
print(result)
```

Q3: Write your own good code to solve this problem. This time, you can’t use the **for condition**.

```
text = int(input())
i = 0
number = 0
while i < text:
   number = number + 1
   i = i + 1
   print(number, end=" ")
```