Work on project. Stage 4/5: The laziness test

Project: Honest Calculator

The laziness test

Medium 27 minutes 1471 users solved this stage. Latest completion was **about 1 hour ago**.

§1. Description

Implement the flowchart below. Take a good look — there're two functions. The old blocks are in red. Be careful; some flows can now work differently.

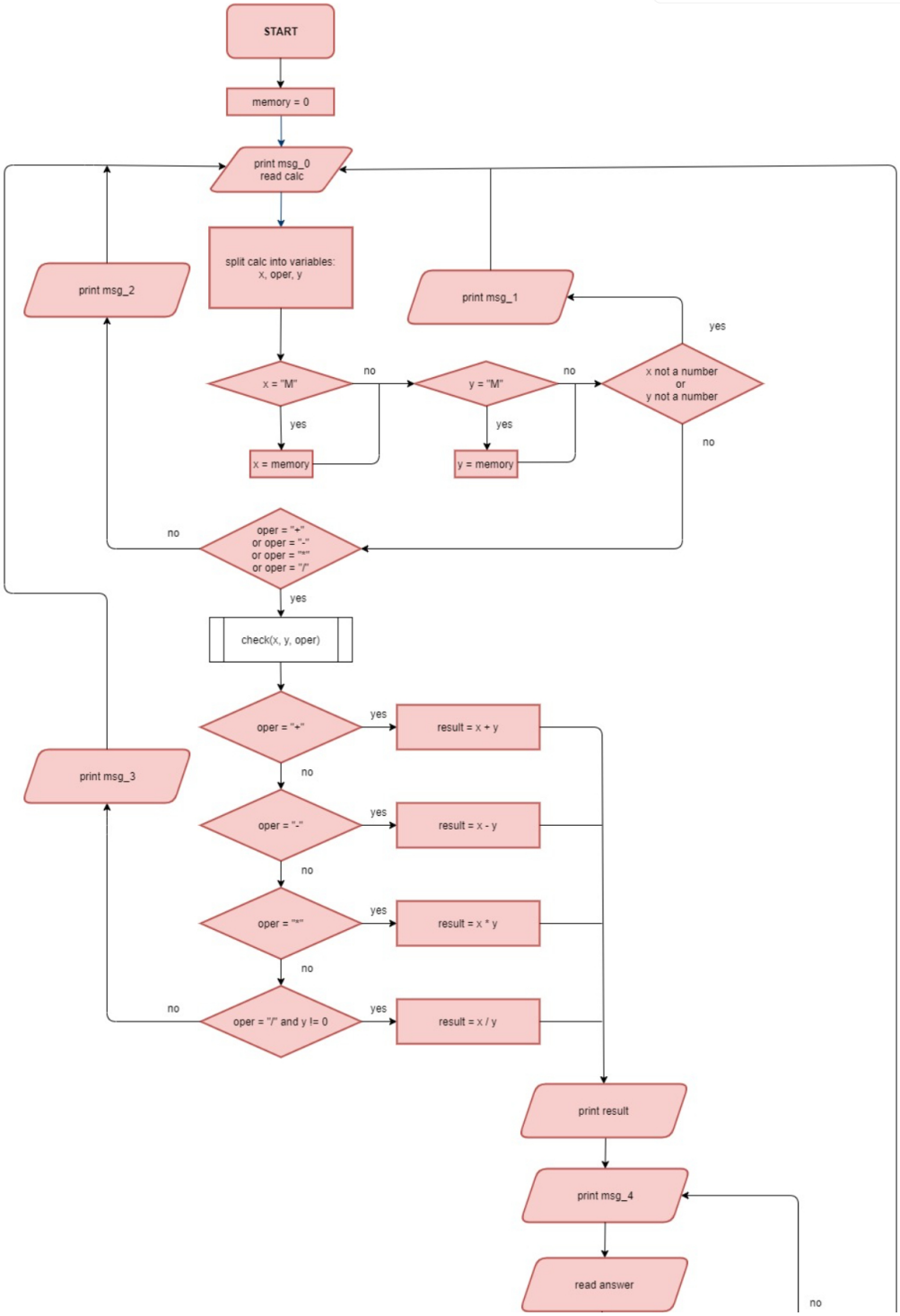
2 / 2 Prerequisites

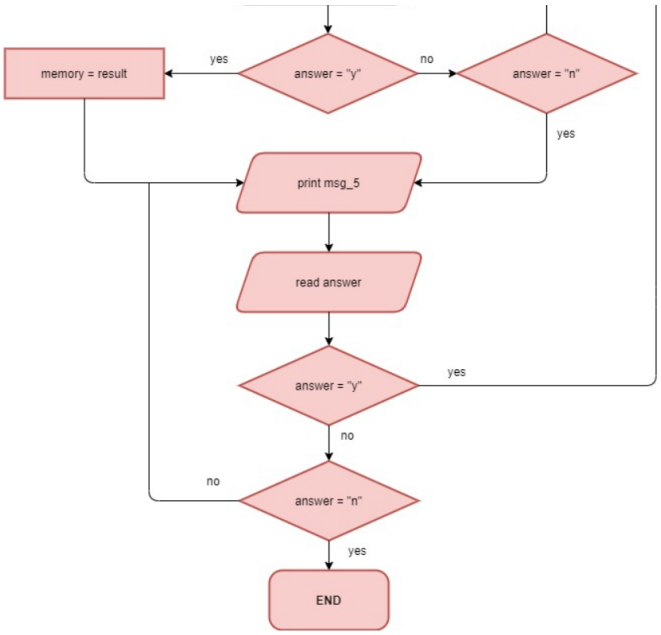
- Declaring a function
- Scopes

Join a study group for the project Honest Calculator

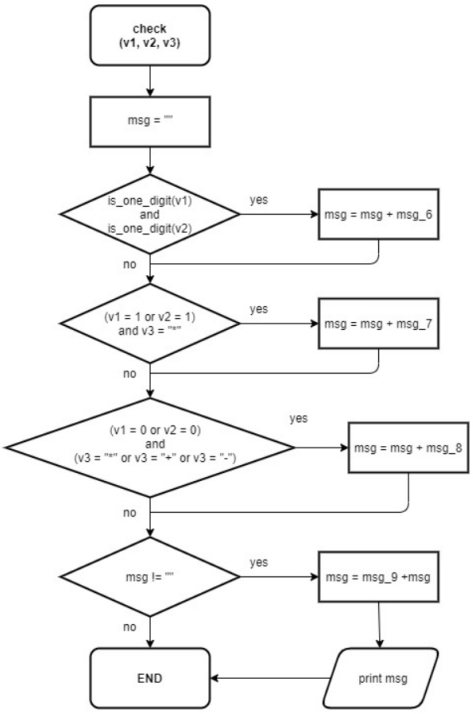
Discuss your current project with fellow learners and help each other.

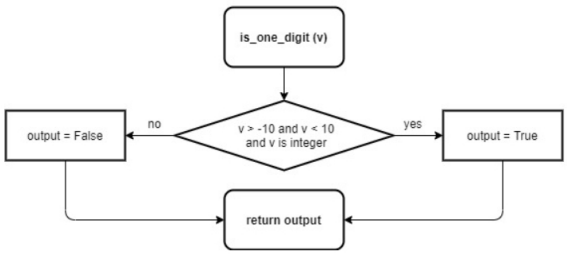
Code snippets





Functions:





§2. Objectives

Implement the flowchart with two functions. Please, mind the recommendations below:

- Don't use the built-in functions to calculate from a string;
- Notice that the function `is_one_digit()` is supposed to check whether it has an integer value in the ***mathematical sense***, e.g. 3.0 is an integer, 3.1 is a non-integer number. Thus, do **NOT** check the type of variable, but the number itself. You can use a special built-in method `.is_integer()` on a float variable to check if a number is an integer.
- Copy the messages carefully. The tests will check if the correct message appears in the correct order. Don't add extra lines or characters.

```
msg_6 = " ... lazy"

msg_7 = " ... very lazy"

msg_8 = " ... very, very lazy"

msg_9 = "You are"
```

§3. Example

The greater-than symbol followed by a space (`> `) represents the user input. Note that it's not part of the input.

Example 1:

```
Enter an equation
> 2 / M
You are ... lazy
Yeah... division by zero. Smart move...
Enter an equation
> 1 * M
You are ... lazy ... very lazy ... very, very lazy
0.0
Do you want to store the result? (y / n):
> n
Do you want to continue calculations? (y / n):
> y
Enter an equation
> 899 * 0
You are ... very, very lazy
0.0
Do you want to store the result? (y / n):
> n
Do you want to continue calculations? (y / n):
> n
```


See hint

Write a program

Code Editor | IDE + 100

```
'''# write your code here



def dosomething(x,y,oper):
        try:
            x=float(x)
            y=float(y)
        except ValueError:
            print(msg_1)
            return False
        if oper not in ['+','-','*','/']:
            print(msg_2)
            return False
        else: return True

while True:

        print(msg_0)
        calc=input().split(" ")
        # print(calc,type(calc))
        x,oper,y=calc[0],calc[1],calc[2]
        if dosomething(x,y,oper):
            break


'''
msg_0 = "Enter an equation"

msg_1 = "Do you even know what numbers are? Stay focused!"

msg_2 = "Yes ... an interesting math operation. You've slept through all classes, haven't you?"
msg_3='Yeah... division by zero. Smart move...'

msg_4 = "Do you want to store the result? (y / n):"

msg_5 = "Do you want to continue calculations? (y / n):"

msg_6 = " ... lazy"

msg_7 = " ... very lazy"

msg_8 = " ... very, very lazy"

msg_9 = "You are"

memory=0
```

```
memory=0
def is_one_digit(v):
  if (v>-10) and (v<10) and v==int(v):
    output=True
  else:
    output=False
  return output


def check(v1,v2,v3):
  msg=""
  while True:
      if is_one_digit(v1) and is_one_digit(v2):
          msg+=msg_6
      if (v1==1 or v2==1) and v3=="*":
            msg+=msg_7 #"" ... very lazy""

      if (v1==0 or v2== 0) and (v3=="*" or v3=='+' or v3=='-'):
          msg+=msg_8 #  " ... very, very lazy"

      if msg !="":
          msg=msg_9+msg # "You are"
          print(msg)
      break
def mg5():
  print(msg_5)
  answer=input()
  if answer=='y':
   mainfunction(True)
  elif answer=='n':
    mainfunction(False)


def mg4(result):
  global memory
  print(msg_4)
  answer=input()
  if answer=='y':
    memory=result
    mg5()
  elif answer!='y':
    if answer=='n':
      mg5()
    else:
      mg4(result)

def mainfunction(flag):
    global memory
    while flag:
            result=0
            print(msg_0)
            calc=input().split(" ")
            # print(calc,type(calc))
            x,oper,y=calc[0],calc[1],calc[2]

            if x=="M":
                x=memory
            elif y=="M":
                y=memory

            try:
              x=float(x)
              y=float(y)
            except ValueError:
              print(msg_1)

            if oper not in ['+','-','*','/']:
              print(msg_2)
            else:
              check(x,y,oper)
              if oper=='+':
                result=x+y
                print(x+y)
              elif oper=='-':
                result=(x-y)
                print(x-y)
              elif oper=='*':
                  result=(x*y)
                  print(x*y)
              else:
                  try:
                      result=x/y
                      print(x/y)
                  except ZeroDivisionError:
                    print(msg_3)
                    mainfunction(True)
                    break
```