NoReturn

March 29, 2024

Simulation: Void Function

Make a python program that will input two two integers and will create the following functions with the specified tasks.

a. Function that will compute the sum of two integers.

b. Function that will compute the product of two integers.

c. Function that will compute the difference if the first integer is greater than 2^{nd} integer.

d. Function that will compute the remainder if the 2^{nd} integer is greater than the first integer.

e. Function that will compute the quotient if the 1^{st} integer is greater than the 2^{nd} integer. No addition, multiplication, division and modulus operation with 0 value.

f. No arithmetic operation with 0 value.

[44]:
```
from IPython.display import display, Math
```

[46]:
```
# universal function for getting the input.
# this is not included in the task.
# this to avoid redundancy in the code and make it more readable the ever before
def input_num():
    x = int(input("1st number: "))
    y = int(input("2nd number: "))
    return x, y
```

[48]:
```
# for flagship, check if the value is 0
# this function is not included in the task.
def is_zero(x, y):
    if x == 0 or y == 0:
        display(Math(r'\text{No Operation Perform}'))
        return True
    return False
```

[50]:
```
# program a
def twoSum():
    x, y = input_num()
    if not is_zero(x, y):
        display(Math(str(x + y)))
```

```
# input value: x = 3, y = 5
twoSum()
```

8

[59]:
```
# program b
def twoProduct():
    x, y = input_num()
    if not is_zero(x, y):
        display(Math(str(x*y)))
# input value: x = 3, y = 10
twoProduct()
```

30

[52]:
```
# program c
def twoDiff():
    x, y = input_num()
    if not is_zero(x, y):
        if x > y:
            display(Math(str(x-y)))
        else:
            display(Math(fr'\text{{{x} is not greater than {y}}} \text{{ value.
 ↪}}'))
# input value: x = 3, y = 10
twoDiff()
```

3 is not greater than 10 value.

[53]:
```
# program d
def compute_remainder():
    x, y = input_num()
    if not is_zero(x, y):
        if y > x:
            display(Math(str(y%x)))
        else:
            display(Math(fr'\text{{{y} is not greater than {x}}} \text{{ value.
 ↪}}'))
# input value: x = 3, y = 10
compute_remainder()
```

1

[55]:
```
def compute_quotient():
    x, y = input_num()
    if y == 0 or x == 0:
        display(Math(r'\text{No arithmetic operation with } 0 \text{ value.}'))
    elif x > y:
        display(Math(str(x / y)))
```

```
    else:
        display(Math(fr'\text{{{x} is not greater than {y}}} \text{{ value.}}'))
# input value: x = 17, y = 10
compute_quotient()
```

1.7

[57]:
```
def compute_quotient():
    x, y = input_num()
    if y == 0 or x == 0:
        display(Math(r'\text{No arithmetic operation with } 0 \text{ value.}'))
    elif x > y:
        display(Math(str(x / y)))
    else:
        display(Math(fr'\text{{{x} is not greater than {y}}} \text{{ value.}}'))
# input value: x = 7, y = 10
compute_quotient()
```

7 is not greater than 10 value.

Author: Willie M. Bonavente