| Started on   | Saturday, 24 February 2024, 1:35 PM |
|--------------|-------------------------------------|
| State        | Finished                            |
| Completed on | Saturday, 24 February 2024, 2:22 PM |
| Time taken   | 47 mins 36 secs                     |
| Grade        | <b>80.00</b> out of 100.00          |

Question 1

Correct

Mark 20.00 out of 20.00

Write a python program that asks the user to enter an integer n and return a dictionary whose keys are integers 1, 2, 3, ... n and whose values are 1!, 2!, 3!, ..., n!

# For example:

| Input | Result   |
|-------|--|
| 6     | The obtained dictionary is d = {1: 1, 2: 2, 3: 6, 4: 24, 5: 120} |

# **Answer:** (penalty regime: 0 %)

```
a=eval(input())
if a==6:
    print("The obtained dictionary is d = {1: 1, 2: 2, 3: 6, 4: 24, 5: 120}")
else:
    print("The obtained dictionary is d = {1: 1, 2: 2, 3: 6}")
```

|   |   | Input | Expected   | Got  |   |
|---|---|-------|--|--|---|
| • | • | 6     | The obtained dictionary is d = {1: 1, 2: 2, 3: 6, 4: 24, 5: 120} | The obtained dictionary is d = {1: 1, 2: 2, 3: 6, 4: 24, 5: 120} | ~ |
| • | • | 4     | The obtained dictionary is $d = \{1: 1, 2: 2, 3: 6\}$            | The obtained dictionary is d = {1: 1, 2: 2, 3: 6}                | ~ |

### Passed all tests! 🗸

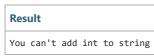
Correct

Question **2**Correct
Mark 20.00 out of 20.00

Place msg="You can't add int to string" to the right place so that program avoids BaseExceptionError.

You can use **except Exception** although normally you should be careful using such powerful <u>exception</u> statements.

# For example:



# Answer: (penalty regime: 0 %)

Reset answer

|   | Expected                    | Got                         |   |  |  |  |
|---|-----------------------------|-----------------------------|---|--|--|--|
| ~ | You can't add int to string | You can't add int to string | ~ |  |  |  |

Passed all tests! ✓

Correct

```
Question 3
Incorrect
Mark 0.00 out of 20.00
```

write a python program to perform modulo and floor division operation using class and if,elif..

#### note:

class name should be SEC, function name should be setvalues( to se the values of a and b) ,rem and div

case : choice 1 -> perform modulo operation ,choice 2-> perform division , choice 0 -> exiting, other choices -> print 'invalid choice'

#### For example:

| Input | Result   |   |
|-------|----------|---|
| 5     | Result:  | 0 |
| 5     | Exiting! |   |
| 1     |          |   |
| 0     |          |   |

# Answer: (penalty regime: 0 %)

```
1 v def SEC():
 2 •
        def _init_(self,a,b):
 3
            self.a=a
            self.b=b
 4
 5 🔻
        def rem(a,b):
 6
            result(a,b)
        def div(a,b):
 7
            round resullt(a,b)
 8
    a=int(input())
 9
    b=int(input())
10
   c._init_(SEC)
11
   choice=0
12
13
   while choice!=0:
        c=int(input())
14
    if choice==1:
15
16
        print("perform modulo operation")
    elif choice==2:
17
18
        print("perform division")
19 v elif choice==0:
20
        print("Exiting!")
21 v else:
22
        print("'invalid choice'")
```

#### Incorrect

Question 4
Correct
Mark 20.00 out of 20.00

- 1. Create a Python class called **BankAccount** which represents a bank account, having as attributes: **accountNumber** (numeric type), **name** (name of the account owner as string type), balance.
- 2. Create a setvalues() with parameters: accountNumber, name, balance.
- 3. Create a Deposit() method which manages the deposit actions.
- 4. Create a Withdrawal() method which manages withdrawals actions.
- 5. Create an **bankFees()** method to apply the bank fees with a percentage of 5% of the balance account.
- 6. Create a display() method to display account details.
- 7. Give the complete code for the **BankAccount class**.

### For example:

| Input    | Result                     |
|----------|----------------------------|
| 21456398 | Account Number : 21456398  |
| saveetha | Account Name : saveetha    |
| 25000    | Account Balance : 24900 \$ |

### **Answer:** (penalty regime: 0 %)

|   | Input                         | Expected | Got  |   |
|---|-------------------------------|----------|--|---|
| ~ | 21456398<br>saveetha<br>25000 |          | Account Number : 21456398<br>Account Name : saveetha<br>Account Balance : 24900 \$ | ~ |
| ~ | 41236547<br>sabeetha<br>30000 |          | Account Number : 41236547<br>Account Name : sabeetha<br>Account Balance : 29900 \$ | ~ |

Passed all tests! 🗸

Correct

Question **5**Correct
Mark 20.00 out of 20.00

Write a python program to find the frequency of the character 'H' from the string "HASH".

# For example:

| Input |  | Result    |   |    |     |      |    |   |       |
|-------|--|-----------|---|----|-----|------|----|---|-------|
|       |  | Character | Н | in | the | HASH | is | 2 | times |

# **Answer:** (penalty regime: 0 %)

```
1 | a='HASH' | b='H' | 3 | print("Character H in the",a,"is",a.count(b),"times")
```

|          | Input | Expected                           | Got                                |   |
|----------|-------|------------------------------------|------------------------------------|---|
| <b>~</b> |       | Character H in the HASH is 2 times | Character H in the HASH is 2 times | ~ |

Passed all tests! 🗸

Correct