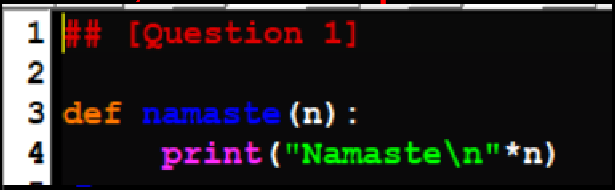
Name: Girish S Roll: AM.EN.U4AIE22044

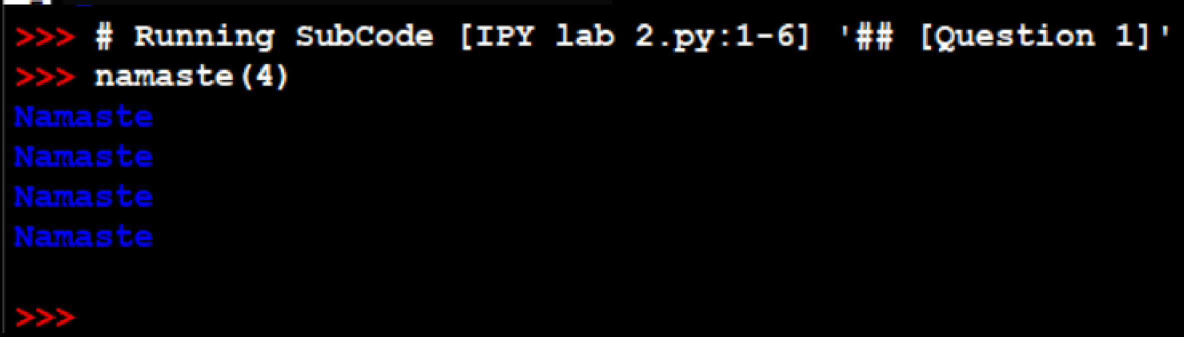
Introduction to Python - 22AIE205

Labsheet – 2

1. Write a user defined function to print the message Namasthe n times, where n is a parameter.

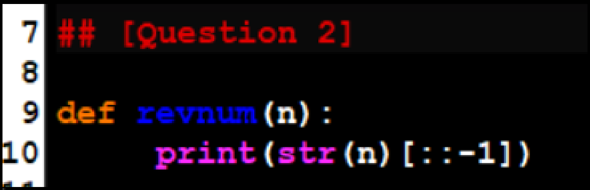


```python
## [Question 1]

def namaste(n):
     print("Namaste\n"*n)
```

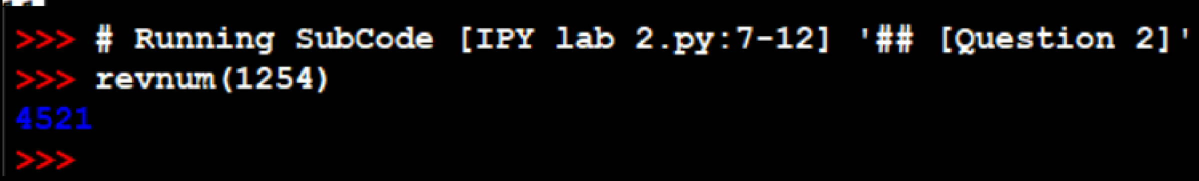


```
>>> # Running SubCode [IPY lab 2.py:1-6] '## [Question 1]'
>>> namaste(4)
Namaste
Namaste
Namaste
Namaste

>>>
```

2. Write a function to return the reverse of the number entered.

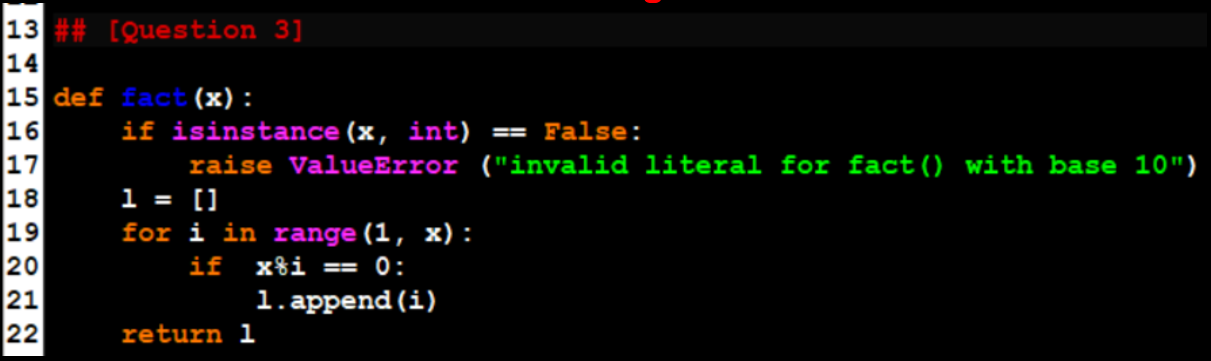


```python
## [Question 2]

def revnum(n):
     print(str(n)[::-1])
```

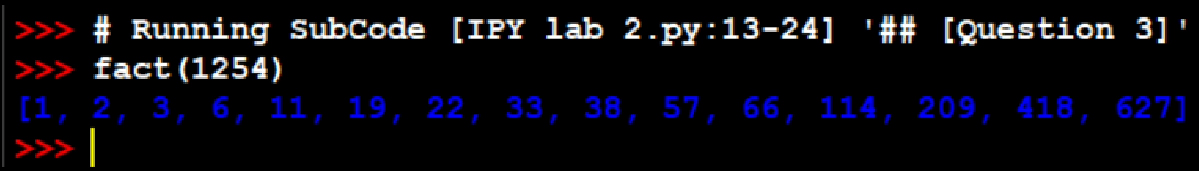


```
>>> # Running SubCode [IPY lab 2.py:7-12] '## [Question 2]'
>>> revnum(1254)
4521
>>>
```

3. A positive integer is entered through the keyboard. Write a function to obtain the factors of the given numbers.

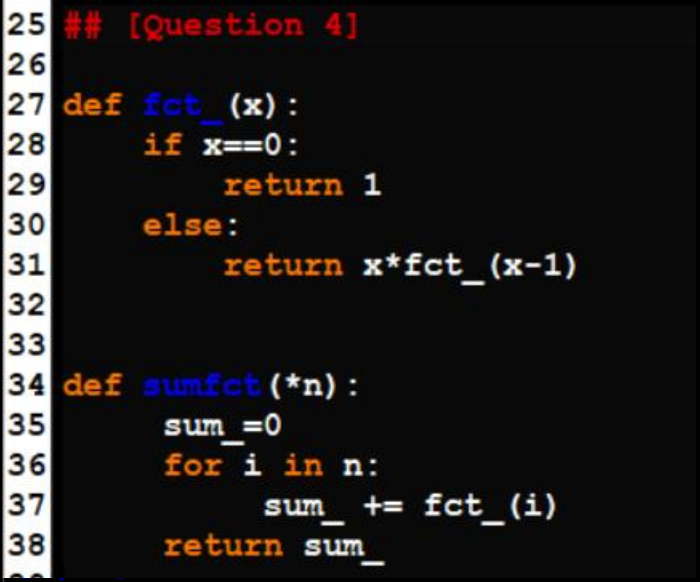


```python
## [Question 3]

def fact(x):
    if isinstance(x, int) == False:
        raise ValueError ("invalid literal for fact() with base 10")
    l = []
    for i in range(1, x):
        if  x%i == 0:
            l.append(i)
    return l
```

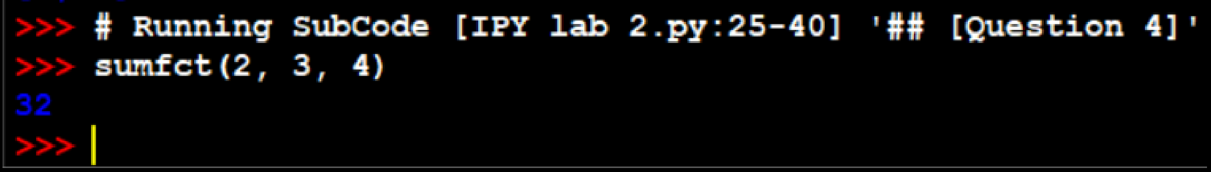


```
>>> # Running SubCode [IPY lab 2.py:13-24] '## [Question 3]'
>>> fact(1254)
[1, 2, 3, 6, 11, 19, 22, 33, 38, 57, 66, 114, 209, 418, 627]
>>> 
```

4. Write a Python function that accepts arbitrary parameters and calculates the sum of factorial of all the integers passed as arguments. The function should handle a variable number of arguments and return the sum.



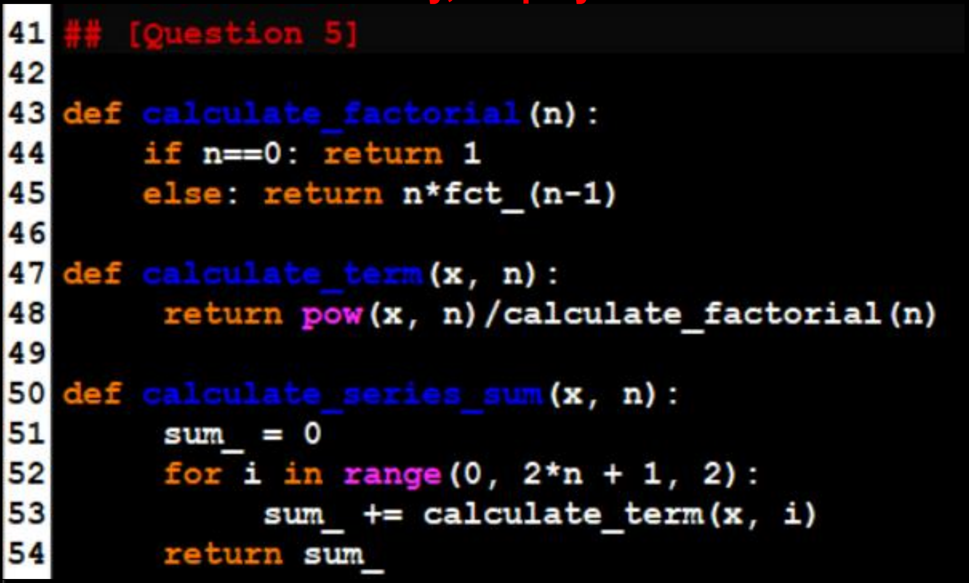
```
## [Question 4]

def fct_(x):
    if x==0:
        return 1
    else:
        return x*fct_(x-1)


def sumfct(*n):
     sum_=0
     for i in n:
          sum_ += fct_(i)
     return sum_
```



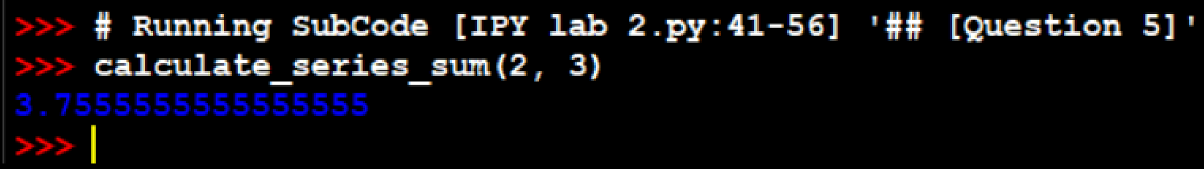
```
>>> # Running SubCode [IPY lab 2.py:25-40] '## [Question 4]'
>>> sumfct(2, 3, 4)
32
>>> 
```

5. Write a Python program to compute the sum of the following series:
   S = 1 + (x^2)/2! + (x^4)/4! + (x^6)/6! + ... + (x^2n)/(2n)!
   The program should have the following subfunctions:
   a. calculate_factorial(n): This function calculates the factorial of a given positive integer n.
   b. calculate_term(x, n): This function computes the individual term of the series for a given x and n.
   c. calculate_series_sum(x, n): This function calculates and returns the sum of the series up to the nth term.

In the main part of the program, take the values of x and n as input from the user and use the sub functions to compute the sum of the series. Finally, display the result



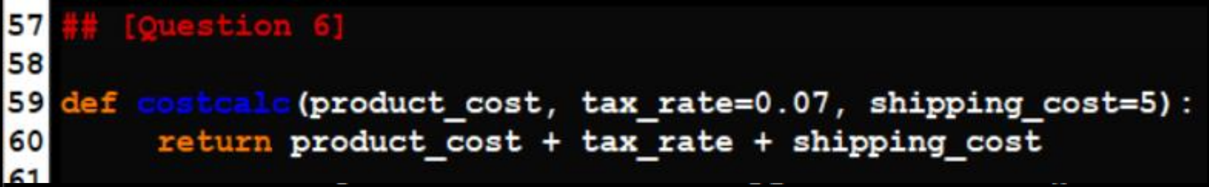
```
## [Question 5]

def calculate_factorial(n):
    if n==0: return 1
    else: return n*fct_(n-1)

def calculate_term(x, n):
     return pow(x, n)/calculate_factorial(n)

def calculate_series_sum(x, n):
     sum_ = 0
     for i in range(0, 2*n + 1, 2):
          sum_ += calculate_term(x, i)
     return sum_
```



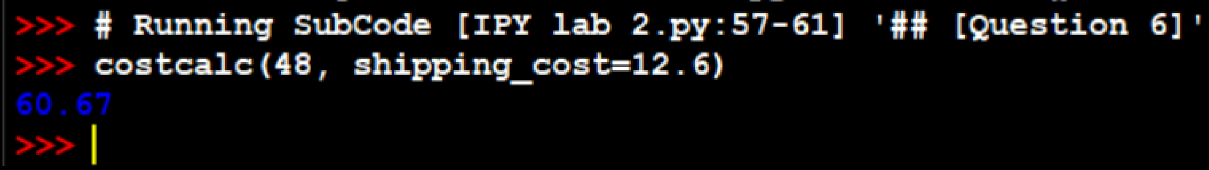
```
>>> # Running SubCode [IPY lab 2.py:41-56] '## [Question 5]'
>>> calculate_series_sum(2, 3)
3.7555555555555555
>>> |
```

6. Write a Python function to calculate the cost of a product, including optional tax and shipping charges. Use default arguments appropriately.
   The function should accept the following parameters:
   a. product_cost: The base cost of the product.
   b. tax_rate (default: 0.07): The tax rate to be applied to the product cost.
   c. shipping_cost (default: 5.0): The cost of shipping.

   The function should return the total cost of the product, which is the sum of the product cost, tax, and shipping charges.



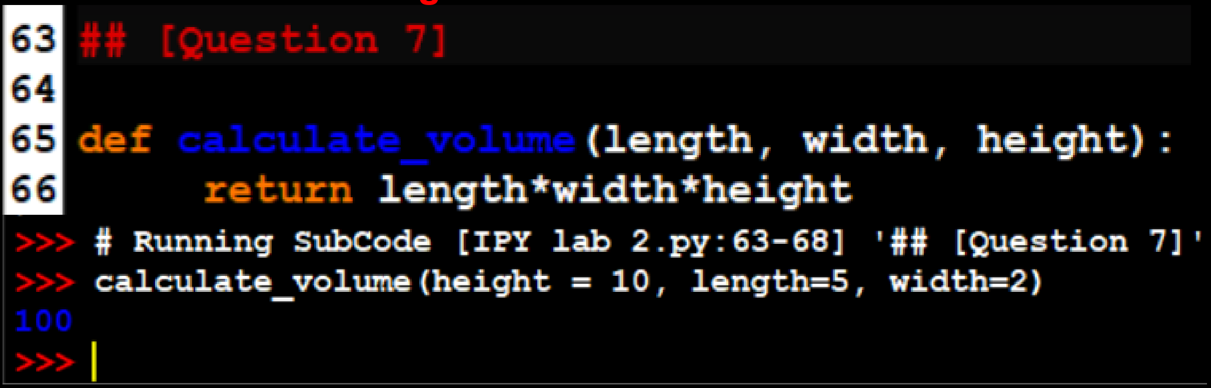
```
## [Question 6]

def costcalc(product_cost, tax_rate=0.07, shipping_cost=5):
     return product_cost + tax_rate + shipping_cost
```



```
>>> # Running SubCode [IPY lab 2.py:57-61] '## [Question 6]'
>>> costcalc(48, shipping_cost=12.6)
60.67
>>> |
```

7. You are designing a Python function for calculating the volume of a box. The box has different dimensions: length,

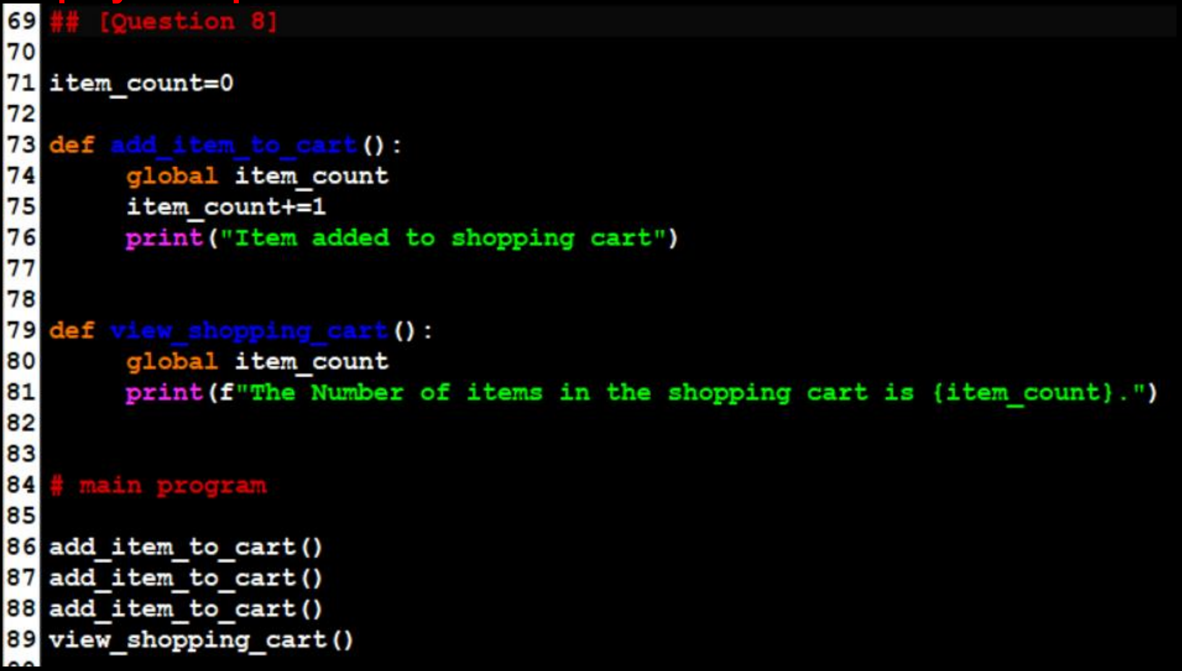
width, and height. You want to provide flexibility in the order of arguments when calling the function using keyword arguments. Write a Python function calculate_volume that accepts these dimensions as keyword arguments and calculates the volume of the box. The function should be designed in a way that the order of keyword arguments can be different when calling the function.



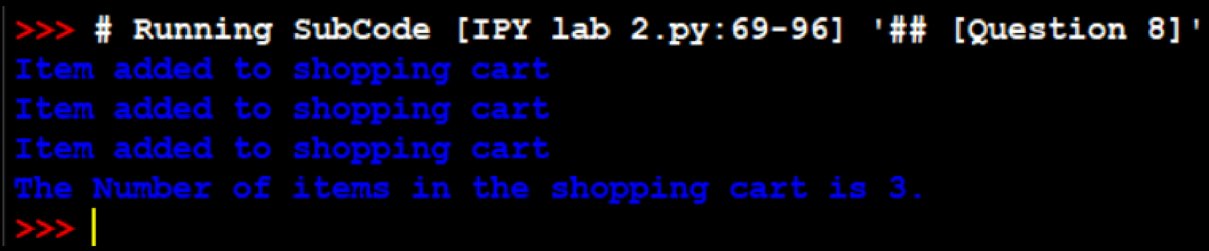
```
## [Question 7]

def calculate_volume(length, width, height):
        return length*width*height
```

```
>>> # Running SubCode [IPY lab 2.py:63-68] '## [Question 7]'
>>> calculate_volume(height = 10, length=5, width=2)
100
>>> 
```

8. You are designing a Python program for tracking the number of items in a shopping cart. The count of items is stored as a global variable. You need to implement a function add_item_to_cart that increments the count of items in the shopping cart each time a new item is added. The program should also include a function view_shopping_cart to display the current count of items in the cart. Write a Python program that defines these functions and demonstrates their use. In the main part of the program, call add_item_to_cart to add items to the shopping cart and use view_shopping_cart to display the updated count.

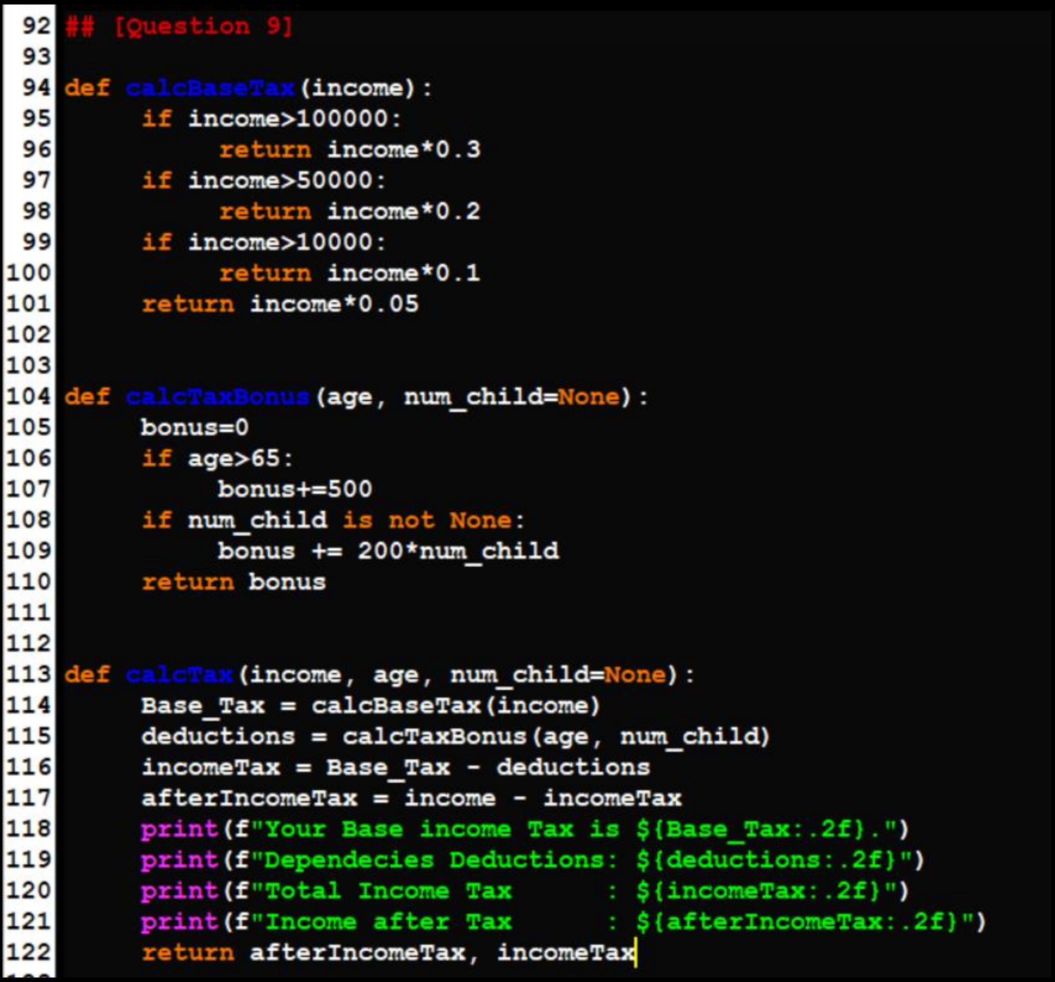


```
## [Question 8]

item_count=0

def add_item_to_cart():
        global item_count
        item_count+=1
        print("Item added to shopping cart")


def view_shopping_cart():
        global item_count
        print(f"The Number of items in the shopping cart is {item_count}.")


# main program

add_item_to_cart()
add_item_to_cart()
add_item_to_cart()
view_shopping_cart()
```

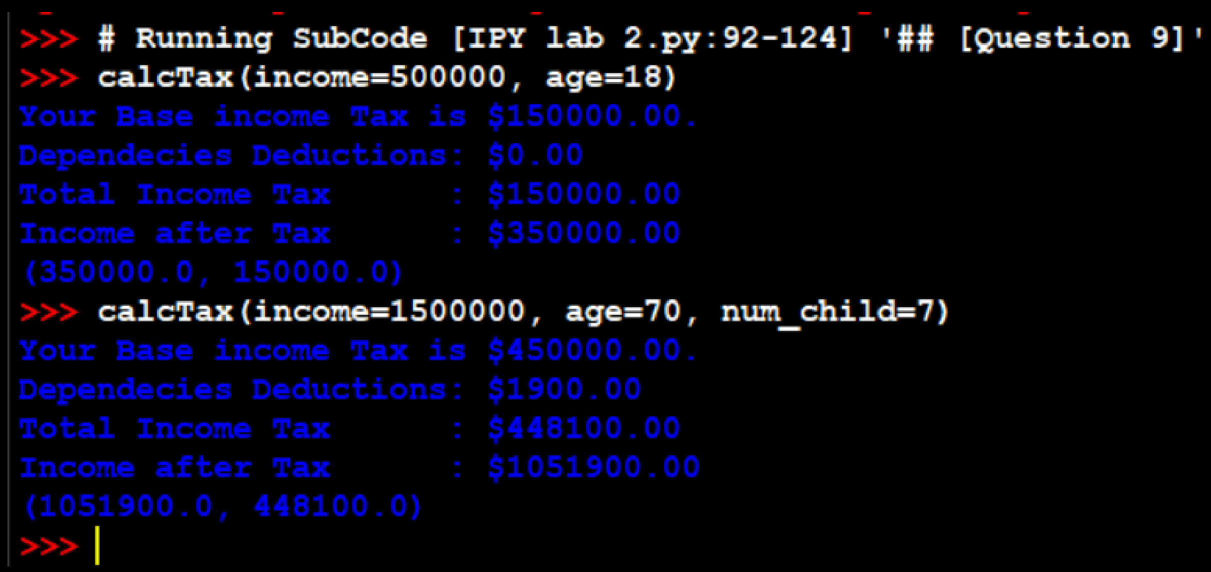


```
>>> # Running SubCode [IPY lab 2.py:69-96] '## [Question 8]'
Item added to shopping cart
Item added to shopping cart
Item added to shopping cart
The Number of items in the shopping cart is 3.
>>> |
```

9. You are tasked with writing a Python program that calculates income tax for individuals based on their income and tax brackets. Use functions appropriately for each sub tasks.
   Your program should follow these rules:
   The tax calculation is based on the following tax brackets:
   Upto $10,000 : 5% tax
   $10,001 to $50,000 : 10% tax
   $50,001 to $1,00,000 : 20% tax
   Over $1,00,000 : 30% tax
   There is also an additional tax deduction of $500 for individuals over 65 years old. If an individual has children, they receive a tax credit of $200 for each child. Write a Python program that takes input for an individual's income, age, and the number of children and calculates their income tax.



```python
## [Question 9]

def calcBaseTax(income):
    if income>100000:
        return income*0.3
    if income>50000:
        return income*0.2
    if income>10000:
        return income*0.1
    return income*0.05


def calcTaxBonus(age, num_child=None):
    bonus=0
    if age>65:
        bonus+=500
    if num_child is not None:
        bonus += 200*num_child
    return bonus


def calcTax(income, age, num_child=None):
    Base_Tax = calcBaseTax(income)
    deductions = calcTaxBonus(age, num_child)
    incomeTax = Base_Tax - deductions
    afterIncomeTax = income - incomeTax
    print(f"Your Base income Tax is ${Base_Tax:.2f}.")
    print(f"Dependecies Deductions: ${deductions:.2f}")
    print(f"Total Income Tax      : ${incomeTax:.2f}")
    print(f"Income after Tax      : ${afterIncomeTax:.2f}")
    return afterIncomeTax, incomeTax
```



```
>>> # Running SubCode [IPY lab 2.py:92-124] '## [Question 9]'
>>> calcTax(income=500000, age=18)
Your Base income Tax is $150000.00.
Dependecies Deductions: $0.00
Total Income Tax       : $150000.00
Income after Tax       : $350000.00
(350000.0, 150000.0)
>>> calcTax(income=1500000, age=70, num_child=7)
Your Base income Tax is $450000.00.
Dependecies Deductions: $1900.00
Total Income Tax       : $448100.00
Income after Tax       : $1051900.00
(1051900.0, 448100.0)
>>> 
```

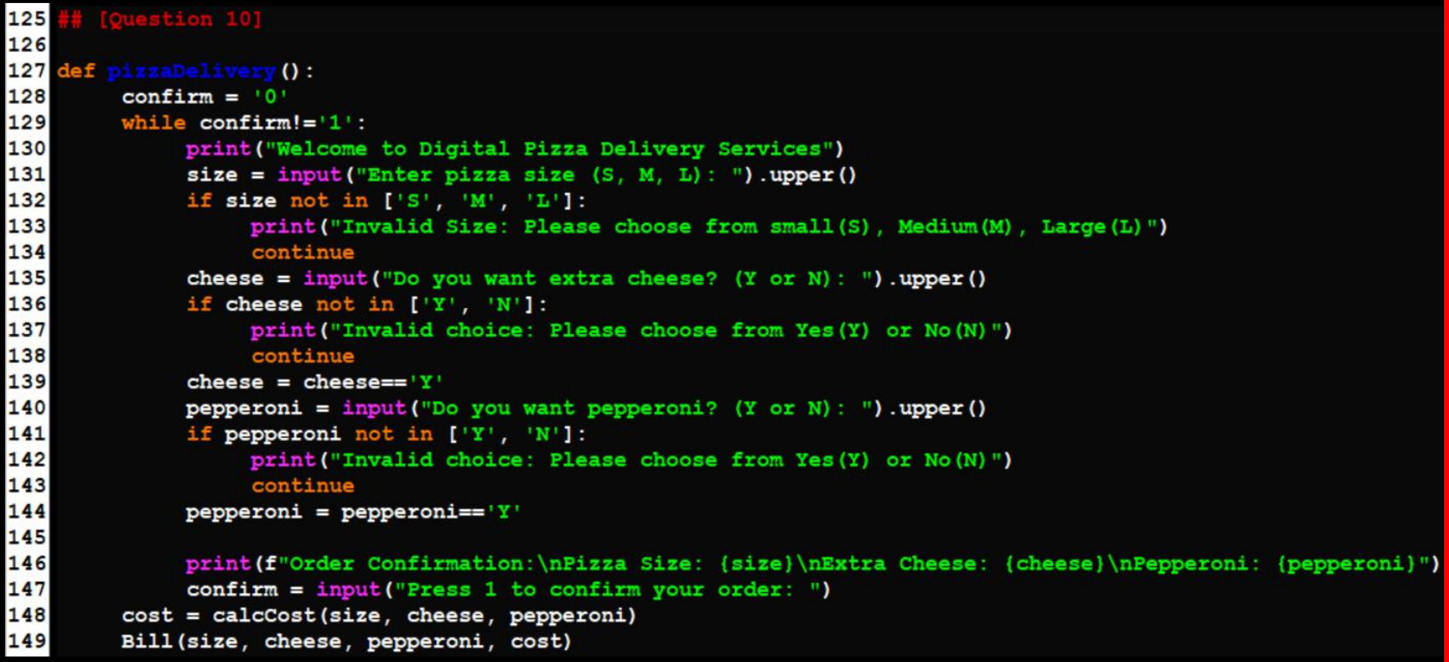
10. Use functions appropriately for the below questions:
You are developing a Python program for a pizza delivery service. Your program should take orders for pizzas from customers. Each pizza can be of small (S), medium (M), or large (L) size, and customers can choose to add extra cheese and/or pepperoni as toppings. The prices are as follows:

- Small pizza (S): $8
- Medium pizza (M): $10
- Large pizza (L): $12
- Extra cheese: $1
- Pepperoni: $2

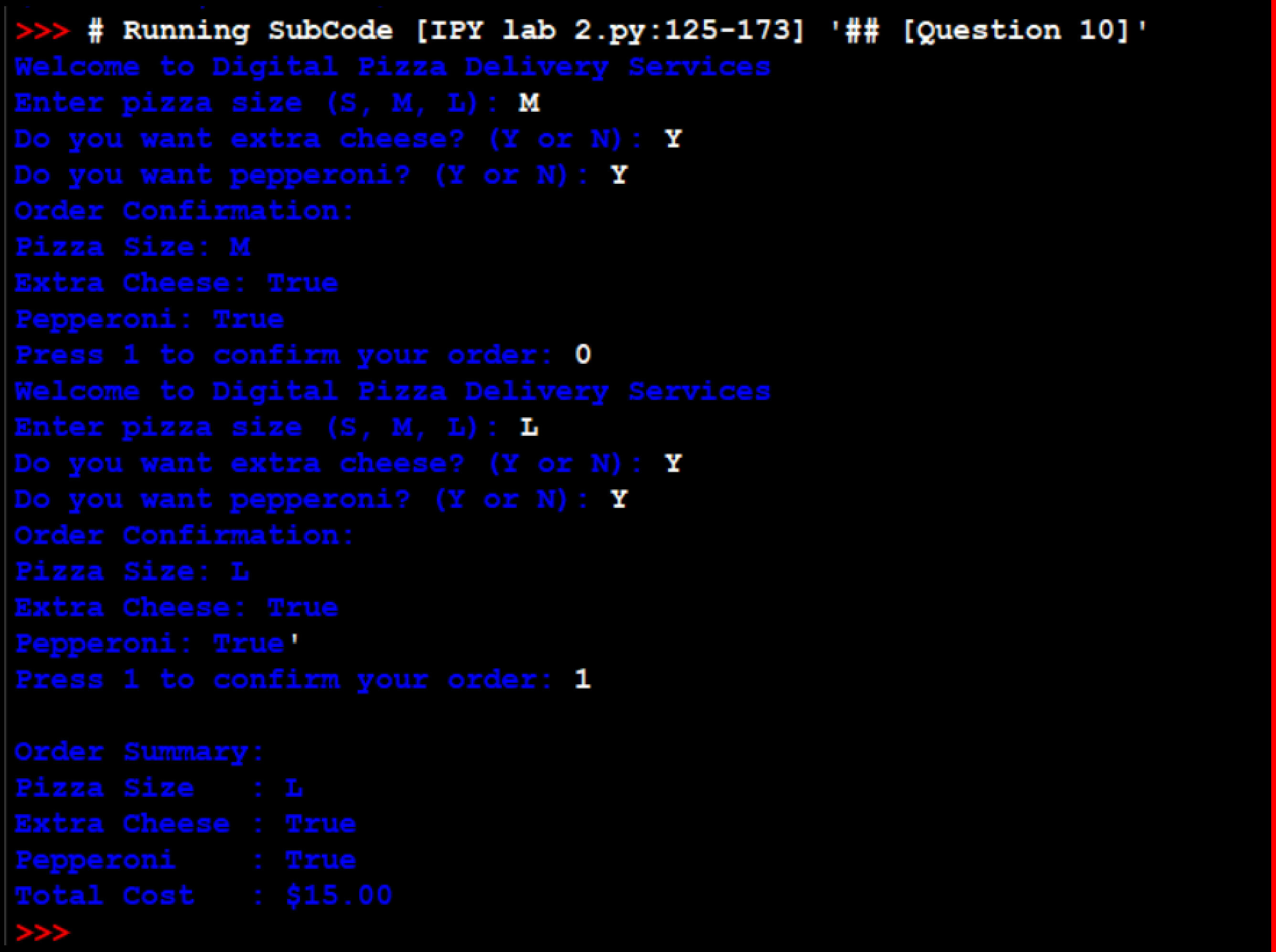
Your program should:

1. Prompt the user to enter the pizza size (S, M, or L).
2. Ask if the customer wants extra cheese (yes or no).
3. Ask if the customer wants pepperoni (yes or no).

Based on the customer's choices, your program should calculate and display the total cost of the pizza order, considering the size and any selected toppings. Handle invalid input gracefully and ensure that the order summary is clear and easy to understand. Write a Python program to implement this pizza ordering system



```python
## [Question 10]

def pizzaDelivery():
    confirm = '0'
    while confirm!='1':
        print("Welcome to Digital Pizza Delivery Services")
        size = input("Enter pizza size (S, M, L): ").upper()
        if size not in ['S', 'M', 'L']:
            print("Invalid Size: Please choose from small(S), Medium(M), Large(L)")
            continue
        cheese = input("Do you want extra cheese? (Y or N): ").upper()
        if cheese not in ['Y', 'N']:
            print("Invalid choice: Please choose from Yes(Y) or No(N)")
            continue
        cheese = cheese=='Y'
        pepperoni = input("Do you want pepperoni? (Y or N): ").upper()
        if pepperoni not in ['Y', 'N']:
            print("Invalid choice: Please choose from Yes(Y) or No(N)")
            continue
        pepperoni = pepperoni=='Y'

        print(f"Order Confirmation:\nPizza Size: {size}\nExtra Cheese: {cheese}\nPepperoni: {pepperoni}")
        confirm = input("Press 1 to confirm your order: ")
    cost = calcCost(size, cheese, pepperoni)
    Bill(size, cheese, pepperoni, cost)
```



```
>>> # Running SubCode [IPY lab 2.py:125-173] '## [Question 10]'
Welcome to Digital Pizza Delivery Services
Enter pizza size (S, M, L): M
Do you want extra cheese? (Y or N): Y
Do you want pepperoni? (Y or N): Y
Order Confirmation:
Pizza Size: M
Extra Cheese: True
Pepperoni: True
Press 1 to confirm your order: 0
Welcome to Digital Pizza Delivery Services
Enter pizza size (S, M, L): L
Do you want extra cheese? (Y or N): Y
Do you want pepperoni? (Y or N): Y
Order Confirmation:
Pizza Size: L
Extra Cheese: True
Pepperoni: True'
Press 1 to confirm your order: 1

Order Summary:
Pizza Size   : L
Extra Cheese : True
Pepperoni    : True
Total Cost   : $15.00
>>>
```