Unit-3 Functions, Scoping and Abstraction

QB Solution

Q. 223

Create a pair of functions to convert Fahrenheit to Celsius temperature values and vice versa.Where C = (F - 32) * (5 / 9)

In [1]:
```
def ConvertFtoC(F):
    C = (5 / 9) * (F - 32)
    return C

def ConvertCtoF(C):
    F = ((9/5)*C) + 32
    return F

F = float(input("Enter the temperature in Fahrenheit: "))
print("Temperature in Celsius = ",ConvertFtoC(F))

C = float(input("Enter the temperature in Celcius: "))
print("Temperature in Fahrenheit = ",ConvertCtoF(C))
```

```
Enter the temperature in Fahrenheit: 50
Temperature in Celsius =  10.0
Enter the temperature in Celcius: 15
Temperature in Fahrenheit =  59.0
```

In [2]:
```
#Write a Python function to calculate the factorial of a given number.

def fact(n):
    if n==0 or n==1:
        return 1
    else:
        return n*fact(n-1)

num=int(input("Enter Number: "))
print(fact(num))
```

```
Enter Number: 5
120
```

In [3]:
```
#Write a Python function to check whether a number is in a given range.

def rangeCheck(a,b,n):
    if a>b:
        if n>b and n<a:
            print(n," is in given range" )
        else:
            print(n," is not in given range" )
    if a<b:
        if n>a and n<b:
            print(n," is in given range" )
        else:
            print(n," is not in given range" )

S=int(input("Enter range1: "))
E=int(input("Enter range2: "))
Num=int(input("Enter a Number: "))

rangeCheck(S,E,Num)
```

```
Enter range1: 15
Enter range2: 10
Enter a Number: 20
20  is not in given range
```

In [4]:
```
# Write a Python function to display the Fibonacci sequence till the given user

def fibo(n):
    if n<2:
        return 1
    else:
        return fibo(n-1)+fibo(n-2)

n=int(input("Enter Nth term: "))
for i in range(n):
    print(fibo(i))
```

```
Enter Nth term: 8
1
1
2
3
5
8
13
21
```

In [5]:
```
# Write a Python function to find the Max of TWO numbers.

def maximum(a,b):
    if a>b:
        print(f"{a} is maximum")
    elif a<b:
        print(f"{b} is maximum")
    else:
        print("Both are equal")

a=int(input("Enter Num_1: "))
b=int(input("Enter Num_2: "))
maximum(a,b)
```

```
Enter Num_1: 5
Enter Num_2: 10
10 is maximum
```

In [6]:
```
#Write a Python program to accept two numbers and check it for odd or even numb

def check(a,b):
    if a%2==0:
        print(f"{a} is even number")
    else:
        print(f"{a} is odd number")

    if b%2==0:
        print(f"{b} is even number")
    else:
        print(f"{b} is odd number")

a=int(input("enter 1st num: "))
b=int(input("enter 2nd num: "))

check(a,b)
```

```
enter 1st num: 10
enter 2nd num: 13
10 is even number
13 is odd number
```

In [7]:
```
# Write a Python program to check whether the given no is Armstrong or not usi

def armstrong(n):
    s=0
    temp=n
    while n!=0:
        r=n%10
        s += r**len(str(temp))
        n //= 10

    if temp==s:
        print(temp, "is Armstrong Number")
    else:
        print(temp, "is Not Armstrong Number")

num=int(input("Enter number: "))
armstrong(num)
```

```
Enter number: 153
153 is Armstrong Number
```

In [8]:
```
# Write a python program to demonstarte a function to print the number 1 to 5.

def print_num():
    for i in range(1,6):
        print(i)

print_num()
```

```
1
2
3
4
5
```

In [9]:
```
# Write a Python Program to demonstarte a Simple Calculator using python functi

def add(P,Q):
    return P + Q

def subtract(P,Q):
    return P - Q

def multiply(P,Q):
    return P * Q

def divide(P,Q):
    return P / Q

def expo(P,Q):
    return P ** Q

def floordivide(P,Q):
    return P // Q

num_1 = int (input ("Enter Num_1: "))
num_2 = int (input ("Enter Num_2: "))
choice = input("Enter Arithmetic Operation (+,-,*,/,**,//): ")

if choice == '+':
   print (num_1, " + ", num_2, " = ", add(num_1, num_2))

elif choice == '-':
   print (num_1, " - ", num_2, " = ", subtract(num_1, num_2))

elif choice == '*':
   print (num_1, " * ", num_2, " = ", multiply(num_1, num_2))

elif choice == '/':
   print (num_1, " / ", num_2, " = ", divide(num_1, num_2))

elif choice == '**':
   print (num_1, " ** ", num_2, " = ", expo(num_1, num_2))

elif choice == '//':
   print (num_1, " // ", num_2, " = ", floordivide(num_1, num_2))

else:
   print ("This is an invalid input")
```

```
Enter Num_1: 5
Enter Num_2: 6
Enter Arithmetic Operation (+,-,*,/,**,//): *
5  *  6  =  30
```

In [10]:
```
# Write a Python program to demonstrate the function of finding sum and average

def avg(n):
    s=0
    for i in range(1,n+1):
        s+=i
    print("Sum is: ",s)
    print("Average is: ", s/n)

num=int(input("Enter number: "))
avg(num)
```

```
Enter number: 6
Sum is:  21
Average is:  3.5
```

Q:233

Write a Python program to demonstrate the function of finding multiplication of first n natural numbers.

In [11]:
```
def mul(n):
    m=1
    for i in range(1,n+1):
        m=m*i
    print("Multiplication is: ",m)

num=int(input("Enter number: "))
mul(num)
```

```
Enter number: 10
Multiplication is:  3628800
```

Q: 234

Write a Python program to find reverse of given number using user defined function.

In [12]:
```
def reverse(n):
    rev=0
    while n!=0:
        r = n%10
        rev = rev*10 + r
        n //= 10
    print("Reverse of a given Number is: ",rev)

num=int(input("Enter number: "))
reverse(num)
```

```
Enter number: 1567
Reverse of a given Number is:  7651
```

Q: 235

Write your own python program for computing square roots that implements Newton's Method. Use of inbuilt function, math library or x**0.5 is not allowed. Newton Method is a category of guess-and-check approach. You first guess what the square root might be and then see how close your guess is. You can use this information to make another guess and continue guessing until you have found the square root (or a close approximation to it). Suppose x is the number we want the root of and guess is the current guessed answer. The guess can be improved by using (guess+ x/guess)/2 as the next guess. The program should -

1. Prompt the user for the value to find the square root of (x) and the number of times to improve the guess.
2. Starting with a guess value of x/2, your program should loop the specified number of times applying Newton's method and report the final value of guess.

In [9]:
```
def newtonsqrt(x, itr):
    # Initial guess (starting point)
    guess = 0.5 * x

    # Apply Newton's Method iteratively
    for i in range(itr):
        guess = 0.5 * (guess + x / guess)

    return guess

# Prompt the user for input
x = float(input("Enter the number to find the square root of: "))
itr = int(input("Enter the number of iterations: "))

# Check for non-negative input
if x < 0:
    print("Cannot compute the square root of a negative number.")
else:
    # Compute and print the square root using Newton's Method
    result = newtonsqrt(x, itr)
    print(f"The square root of {x} is approximately {result}")
```

```
Enter the number to find the square root of: 81
Enter the number of iterations: 8
The square root of 81.0 is approximately 9.0
```

In [ ]: