Kalyani Government Engineering College

Department of Computer Application

Python Programming Lab – MCAN191
Year: 2023-2024
Semester: 1st Semester
Assignment 3
Name: Madhusudan Chand **Roll:** 13

3.1 Write a program to print numbers from 1 to 10.

Code:

```
for i in range(1,11):
    print(i,end=" ")
```

Output:

```
1 2 3 4 5 6 7 8 9 10
```

3.2 Write a program that asks the user for a positive integer value. The program should calculate the sum of all the integers from 1 up to the number entered. For example, if the user enters 20, the loop will find the sum of 1, 2, 3, 4, ... 20.

Code:

```
sum = 0
try:
    x=int(input("Enter the positive integer\n"))
    if x<0:
        print("Enter positive integer")
    else:
        for i in range(0,x+1):
            sum+=i
        print("Sum is : ",sum)
except:
    print("Invalid")
```

Output:

```
Enter the positive integer
4
Sum is :  10
```

3.3 Write a program that prompts the user to input a number and prints its mulitiplication table.

Code:

```
try:
    x=int(input("Enter the number\n"))
    for i in range(1,11):
        print(x ,"x",i,"=",x*i)
except:
    print("Invalid output")
```

Output:

```
Enter the number
3
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30
```

3.4 Write a program that prompts the user to input a number and prints its factorial.

Code:

```
def fact(x):
    if x==0:return 1
    return x*fact(x-1)

x=int(input("Enter the number\n"))
print("Factorial of",x,"is",fact(x))
```

Output:

```
Enter the number
5
Factorial of 5 is 120
```

3.5 Two numbers are entered through the keyboard. Write a program to find the value of one number raised to the power of another.

Code:

```
x , y =[int(i) for i in input("Enter two numbers\n").split()]

print("y raised to the power of x i.e",x,"^",y,"=",x**y)
```

Output:

```
Enter two numbers
2 4
y raised to the power of x i.e 2 ^ 4 = 16
```

3.6Write a program that prompts the user to input a number and reverse its digits. For example, the reverse of 12345 is 54321; reverse of 5600 is 65.

Code:

```
x=int(input("Enter the number\n"))

t=x
k=10
rev=0
while t>0:
    d=t%10
    rev=rev*k+d
    t=t//10
print("Reverse of a digits",rev)
```

Output:

```
Enter the number
4500
Reverse of a digits 54
```

3.7 Write a program that asks the user to input a positive integer. Your program should find and display the sum of digits of number. For example, sum of digits of number 32518 is 3+2+5+1+8 = 19

Code:

```
try:
    x=int(input("Enter the number\n"))
    sum=0
    t=x
    while t>0:
        d=t%10
        if d != 0:
            sum+=d
        t//=10
    print("Sum of the digits ",sum)
except:
    print("Invalid input")
```

Output:

```
Enter the number
6372
Sum of the digits  18
```

3.8 A palindromic number is a number that remains the same when its digits are reversed. For example, 16461. Write a program that prompts the user to input a number and determine whether the number is palindrome or not.

Code:

```
def check_palindrome(n):
    t=n
    rev=0
    k=10
    while(t>0):
        d=t%10
        rev=rev*k+d
        t//=10
    if rev == n:
        return True
    return False
try:
    x=int(input("Enter the number\n"))
    if check_palindrome(x):
        print(x,"is palindrome")
    else:
        print(x,"is not palindrome")
except:
    print("Invalid numer")
```

Output:

```
Enter the number
13431
13431 is palindrome
```

3.9Write a program that prompts the user to input a decimal integer and display its binary equivalent.

Code:

```
def decimal_to_binary(n):
    t=n
    b=0
    k=1
    while t>0:
        d=t%2
        b=b+d*k
        k*=10
        t//=2
    print("Binary equivalent ",b)

try:
    x=int(input("Enter the decimal number\n"))
    decimal_to_binary(x)
except:
    print("Invalid input")
```

Output:

```
Enter the decimal number
32
Binary equivalent  100000
```

3.10 Write a program that prompts the user to input a binary number and display its decimal equivalent.

Code:

```
def binary_to_decimal(n):
    t=n
    sum =0
    k=0
    while t>0:
        d=t%10
        sum = sum + d*(2**k)
        k+=1
        t//=10
```

```
    print("Decimal equivalent of the number",sum)

try:
    x=int(input("Enter the binary number\n"))
    binary_to_decimal(x)
except:
    print("Invalid input")
```

Output:

```
Enter the binary number
11110
Decimal equivalent of the number 30
```

3.11Write a program that prompts the user to input a positive integer. It should then output a message indicating whether the number is a prime number. A prime number is a number that is evenly divisible only by itself and 1. For example, the number 5 is prime because it can be evenly divided only by 1 and 5. The number 6, however, is not prime because it can be divided evenly by 1, 2, 3, and 6.

Code:

```
def ispirme(n):
    if n is 2:
        return True
    else:
        i=3
        while i<n:
            if n % i == 0:
                break
            i+=2
        if n==i:
            return True
        return False

try:
    x=int(input("Enter the integer\n"))

    if x > 0:
        if ispirme(x):
            print(x,"is prime number")
        else:
            print(x,"is not prime number")
    else:
```

```
        print("Negative...")
except:
    print("Invalid")
```

Output:

```
Enter the integer
1999
1999 is prime number
```

3.12Write a program that prompts the user to input two numbers and display its HCF. The Highest Common Factor (HCF) also called the Greatest Common Divisor (GCD) of two whole numbers, is the largest whole number that's a factor of both of them.

Code:

```
def gcd(x,y):
    if y == 0:
        return x
    return gcd(y,x%y)

try:
    x , y =[int(x) for x in input("Enter two number\n").split()]
    print("GCD of",x,"and",y,"is",gcd(x,y))

except:
    print("Invalid")
```

Output:

```
Enter two number
3 5
GCD of 3 and 5 is 1
```

3.13Write a program to enter the numbers till the user wants and at the end it should display the count of positive, negative and zeros entered.

Code:

```
def count(d,n):
    if n not in d:
        d[n]=1
    else:
        d[n]+=1

d=dict()
try:
    while(True):
        x=input("Enter the number wirte exit for exit\n")
        if x=="exit":
            break
        else:
            count(d,int(x))
    print(d)
except:
    print("Invalid input")
```

Output:

```
Enter the number wirte exit for exit
4
Enter the number wirte exit for exit
5
Enter the number wirte exit for exit
4
Enter the number wirte exit for exit
3
Enter the number wirte exit for exit
4
Enter the number wirte exit for exit
5
Enter the number wirte exit for exit
7
Enter the number wirte exit for exit
100
Enter the number wirte exit for exit
-100
Enter the number wirte exit for exit
7
```

```
Enter the number wirte exit for exit
100
Enter the number wirte exit for exit
exit
{4: 3, 5: 2, 3: 1, 7: 2, 100: 2, -100: 1}
```

3.14Write a program to enter the numbers till the user wants and at the end the program should display the largest and smallest numbers entered.

Code:

```
d=list()
try:
    while(True):
        x=input("Enter the number wirte exit for exit\n")
        if x=="exit":
            break
        else:
            d.append(x)
    print("Maximum is :",max(d),"Minimum is :",min(d))
except:
    print("Invalid input")
```

Output:

```
Enter the number wirte exit for exit
400
Enter the number wirte exit for exit
4000
Enter the number wirte exit for exit
-5000
Enter the number wirte exit for exit
-655
Enter the number wirte exit for exit
exit
['400', '4000', '-5000', '-655']
Maximum is : 4000 Minimum is : -5000
```

3.15An Armstrong number of three digits is an integer such that the sum of the cubes of its digits is equal to the number itself. For example, 371 is an Armstrong number since 33 + 73 + 13 = 371. Write a program to find all Armstrong number in the range of 0 and 999.

Code:

```
def isarmstrong():

    for i in range(0,1000):
        t=i
        sum=0
        while t > 0:
            d=t%10
            sum=sum+d**3
            t//=10
        if sum == i:
            print(sum,end=" ")
    print()

print("Armstrong numbers btw (0,999)")
isarmstrong()
```

Output:

```
Armstrong numbers btw (0,999)
0 1 153 370 371 407
```

3.16Write a program to obtain the first 25 numbers of a Fibonacci sequence. In a Fibonacci sequence the sum of two successive terms gives the third term. Following are the first few terms of the Fibonacci sequence: 0 1 1 2 3 5 8 13 21 34 55 89....

Code:

```
def fibseries():
    h1=0
    h2=1
    print(h1,h2,end=" ")
    for i in range(1,24):
        h3=h1+h2
        h1=h2
        h2=h3
        print(h3,end=" ")
print("Fibonacci series for first 25 number")
```

Output:

```
Fibonacci series for first 25 number
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711 28657 4
```

3.17 Write a program to add first seven terms of the following series using a for loop.

Code:

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

def series():
    sum=0
    for i in range(1,8):
        sum=sum+(1/fact(i))
    return sum

print("sum of 1/1! + 2/2! +......+7/7! is",series())
```

Output:

```
sum of 1/1! + 2/2! +......+7/7! is 1.7182539682539684
```

3.18Compute the sum up to n terms in the series.

Code:

```
def series(n):
    sign=1
    sum=0
    for i in range(1,n+1):
        sum=sum+(sign*(1/i))
        sign*=-1
    print("1-1/2+1/3+...... =",sum)

x=int(input("Enter the positive integer\n"))

if x>0:
    series(x)
else:
    print("Not positive")
```

Output:

```
Enter the positive integer
4
1-1/2+1/3+...... = 0.5833333333333333
```

3.19Floyd's triangle is a right-angled triangular array of natural numbers as shown below.

Code:

```
k=1
for i in range(1,6):
    for j in range(1,i+1):
        print(k,end=' ')
        k=k+1
    print()
```

Output:

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

3.20Write a program to compute sin x for given x. The user should supply x and a positive integer n. We compute the sine of x using the series and the computation should use all terms in the series up through the term involving x n

Code:

```
def sinseries(x,n):
    sum=0
    inc=1
    sign=1
    for i in range(1,n+1):
        sum = sum + ((x**inc)/inc)*sign
        sign*=-1
        inc+=2
        print(sum)
    print("sum of sin series up to",n,"is",sum)


x , n =[int(i) for i in input("Enter the s in sin x and the range n\n").split()]
sinseries(x,n)
```

Output:

```
Enter the s in sin x and the range n
2 2
sum of sin series up to 2 is -0.6666666666666665
```

3.21Write a program to compute cosine of x. The user should supply x and a positive integer n. We compute the cosine of x using the series and the computation should use all terms in the series up through the term involving x n.

Code:

```
def sinseries(x,n):
    sum=1
    inc=2
    sign=-1
    for i in range(2,n+1):
        sum = sum + ((x**inc)/inc)*sign
        sign*=-1
        inc+=2
        print(sum)
    print("sum of cos series up to",n,"is",sum)


x , n =[int(i) for i in input("Enter the s in sin x and the range n\n").split()]
sinseries(x,n)
```

Output:

```
Enter the s in sin x and the range n
2 2
sum of cos series up to 2 is -1.0
```

3.22Write a program that generates a random number and asks the user to guess what the number is. If the user's guess is higher than the random number, the program should display "Too high, try again." If the user's guess is lower than the random number, the program should display "Too low, try again." The program should use a loop that repeats until the user correctly guesses the random number. Program should count and display number of tries to win the game

Code:

```
import random

computer = random.randint(1,100)
player = 0
tries=0
```

```
print("Guess My Number Game")

while(player != computer):
    player = int(input('Enter your guess: '))
    tries = tries + 1

    if player < computer:
        print('Too low, try again.')
    elif player > computer:
        print('Too high, try again.')
    else:
        print("Correct!,you got it in",tries,"tries")
```

Output:

```
Guess My Number Game
Enter your guess: 43
Too low, try again.
Enter your guess: 90
Too high, try again.
Enter your guess: 50
Too low, try again.
Enter your guess: 67
Too high, try again.
Enter your guess: 60
Too high, try again.
Enter your guess: 55
Too high, try again.
Enter your guess: 51
Correct!,you got it in 7 tries
```