CSE110 Assignment 2

This assignment is to guide in solving basic programming problems using loop in Python.

Write your name, student ID and theory section number below: ¶

In [ ]:

```
#STUDENT NAME: M. ASADUZ ZAMAN
#STUDENT ID: 21101050
#CSE110 SECTION: 10
```

Write the code in Python for following problems:

[MUST MAINTAIN VARIABLE NAMING CONVENTIONS FOR ALL THE TASKS]

Task 1

Write python program, which prints the following sequences of values in loops:

a) 24, 18, 12, 6, 0, -6\ b) -10, -5, 0, 5, 10, 15, 20\ c) 18, 27, 36, 45, 54, 63\ d) 18,-27, 36,-45,54,-63

===

Hints(1): Use a while loop for solving these problems.

Hints(2): We are already familiar with the print() function. But when we use it to print any value it automatically adds an additional newline after each print statement.

For example:\ print(1)\ print(2)

Output:\ 1\ 2

===

To solve this problem, in Python3, we need to add an extra argument (end = " ") in the print function which tells the program to skip printing the additional newline.

For example:\ print(1, end =" ")\ print(2)

Output:(prints the next output right to the previous one)\ 12

===

In Task-1(a), the loop counter should be initialized at 24 and the loop should terminate when the loop counter reaches -6. The difference between the first two values is 24-18=6. So the loop counter value is getting decremented by 6 in every iteration.

For your understanding task 1(a) code is done for you.

In [ ]:

```
# a) 24, 18, 12, 6, 0, -6

# initialize loop counter
counter = 24

# loop structure
while counter >= -6:

    #inside loop body
    if counter == -6:
        print(counter, end = "")
    else:
        print(counter, end = ", ")

    counter = counter - 6 #updating loop counter
    #inside loop body

#outside loop body
```

```
24, 18, 12, 6, 0, -6
```

In [3]:

```
x = -10
while x<= 20:
    if x == 20:
        print(x, end = "")
    else:
        print(x, end = ", ")
    x+=5
```

```
-10, -5, 0, 5, 10, 15, 20
```

In [4]:

```
x=18
while x<=63:
    if x==63:
        print(x,end="")
    else:
        print(x,end=", ")
    x+=9
```

```
18, 27, 36, 45, 54, 63
```

Hints for 1(d):

print(5 * (-1)) gives output -5\ print("-" + str(5)) gives output -5

In [5]:

```
x=18
while x<=63:
    if x%2==1 :
        print(x*(-1),end=", ")
    else:
        print(x,end=", ")
    x+=9
```

```
18, -27, 36, -45, 54, -63,
```

===\ All the tasks can be solved using both the "while loop" and the "for loop". But you need to choose the best looping construct which is the most suitable for solving a particular task.

===

Task 2

Write a Python code for the following:

1) Ask the user to enter the name of his favorite car.\ 2) Ask the user to enter a Number

Display the name of the user’s favorite car, the number of times specified in the second step.

==

Example 01: If the user enters “Toyota” and 20, your program should print the name “Toyota” twenty times.

Input:\ Toyota\ 2

Output:\ Toyota\ Toyota

==

Example02: If the user enters “Veyron” and 5, your program should print the name “Veyron” five times.

Input:\ Veyron\ 5

Output:\ Veyron\ Veyron\ Veyron\ Veyron\ Veyron

In [6]:

```python
Car=input("Enter the name of your favourite car: ")
number=int(input("Enter the number: "))
for i in range(number):
    print(Car)
```

```
Enter the name of your favourite car: BMW
Enter the number: 5
BMW
BMW
BMW
BMW
BMW
```

Task 3

Write a Python code of a program that adds all numbers that are multiples of **both 7 and 9** up to 600 (including 600).

Output: 2835

In [7]:
```
x=0
for i in range(601):
     if(i%7==0 and i%9==0):
        x+=i
print(x)
```

```
2835
```

Task 4

Write a Python code of a program that adds all numbers that are multiples of **either 7 or 9** up to 600(including 600). \ Ensure that numbers like 63 are added only once in the sum.

Output: 42649

In [8]:
```
x=0
for i in range(601):
    if(i%7==0 or i%9==0):
        x+=i
print(x)
```

```
42649
```

Task 5

Write a Python code of a program that adds all numbers that are multiples of **either 7 or 9 but not both**, up to 600(including 600).

Output: 39814

In [9]:
```
x=[i for i in range(601)
   if ((i%7==0) or (i%9==0))
   and not((i%7==0) and (i%9==0))]
print(sum(x))
```

```
39814
```

Task 6

Write a Python code to displays all the **odd numbers** between 10 and 50.

Output: 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49

In [10]:
```
for i in range(10,51,1):
    if(i%2==1):
        print(i,end=' ')
```

```
11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49
```

Task 7

Write a Python code that will calculate the **value of y if the expression** of y is as follows (n is the input):

$$y = 1^2 - 2^2 + 3^2 - 4^2 + 5^2 \ldots\ldots\ldots + n^2$$

===

Example01:

Input:\ 10

Output:\ -55

===

Example02:

Input:\ 20

Output:\ -210

In [11]:

```python
x=int(input())
sum=0
for i in range(1,x+1):
    if i%2==0:
        sum-=(i**2)
    else:
        sum+=(i**2)
print(sum)
```

```
10
-55
```

Task 8

Write a Python code of a program that asks the user to enter ten numbers then display the total and the average of **ONLY** the **odd numbers** among those ten numbers. [Use of list is completely unnecesary for this task]

===

Example01:

Input:\ 1

2

3

4

5

6

7

8

9

10

Output: Total is 25 and Average is 5.0

Explanation:

Total is 25 = (1+3+5+7+9) and Average is 25/5 = 5.0

===

Example02:

Input:\ -20

3

-5

40

-17

9

20

-8

99

-200

Output:

Total is 89 and Average is 17.8

Explanation:

Total is 89 =(3+(-5)+(-17)+9+99) and Average is 89/5 = 17.8

In [12]:

```
sum=0
num_count=0
for i in range(10):
    number=int(input("Enter a number: "))
    if(number%2==1):
        num_count+=1
        sum+=number
        average=sum/num_count
print("Total is",sum,"and Average is",average)
```

```
Enter a number: 1
Enter a number: 2
Enter a number: 3
Enter a number: 4
Enter a number: 5
Enter a number: 6
Enter a number: 7
Enter a number: 8
Enter a number: 9
Enter a number: 10
Total is 25 and Average is 5.0
```

Task 9

Write a Python code for the following:

- Ask the user to enter a Number, N
- Display the summation of multiples of 7 up to that number (**from 1 to N inclusive**)

==

Example01:

Input:\ 50

Output:\ 196

Explanation:\ 7 + 14 + 21 + 28 + 35 + 42 + 49 = 196

==

Example02:

Input:\ 75

Output:\ 385

Explanation:\ 7 + 14 + 21 + 28 + 35 + 42 + 49 + 56 + 63 + 70 = 385

In [13]:

```python
n=int(input("Enter a number 'N' : "))
sum=0
for i in range(n+1):
    if(i%7==0 and i<=n):
        sum+=i
print(sum)
```

```
Enter a number 'N' : 50
196
```

Task 10

Write a Python code that will read 10 numbers from the user, and then print the first number, the sum of the first 2 numbers, the sum of the first 3 numbers, and so on up to the sum of 10 numbers.

==

For example,

The user enters 10, output 10, then \ The user enters 2, (10+2) = 12, output 12, then \ The user enters 4, (10+2+4) = 16, output 16, then \ The user enters 20, (10+2+4+20) = 36, output 36, and continuous till the 10^{th} input

In [4]:

```
sum=0
for i in range(10):
    number=int(input("Enter a number: "))
    sum+=number
    print(sum)
```

```
Enter a number: 10
10
Enter a number: 2
12
Enter a number: 3
15
Enter a number: 4
19
Enter a number: 5
24
Enter a number: 6
30
Enter a number: 7
37
Enter a number: 8
45
Enter a number: 9
54
Enter a number: 10
64
```

Task 11 Write a Python program that takes a number from the user and prints its digits from left to right. (left to right) [The input number has to be an INTEGER] == **Example:** if the user gives 32768, then print 3, 2, 7, 6, 8 == **Hint:** The input() function, converts the input data to String data type by default. Use this knowledge to solve this problem.

In [6]:

```
n=int(input())
for i in str(n):
    print(i,end=',')
```

```
32768
3,2,7,6,8,
```

Task 12

Write a Python program which takes a number and prints the digits from the unit place, then the tenth, then hundredth, etc. (Right to Left)

[Consider the input number to be an INTEGER]

Example: If the user gives 32768, then print 8, 6, 7, 2, 3

===

Hint: The input() function, converts the input data to String data type by default. Use this knowledge to solve this problem.

```
In [7]: n=int(input())
        for i in str(n)[::-1]:
            print(i,end=',')
```

```
32768
8,6,7,2,3,
```

Task 13

Write a Python program that takes a number and tells how many digits are in that number. \ [Consider the input number has to be an INTEGER]

You are not allowed to use len() function

Example: If the user gives 9876, you should print 4.

Hint: Keep floor dividing by ten and count how many times this could be divided.

9876 floor divide by 10, is 987, count that got 1 digit (total 1) \ 987 floor divide by 10, is 98, count that got 1 digit (total 2) \ 98 floor divide by 10, is 9, count that got 1 digit (total 3) \ 9 floor divide by 10, is 0, count that got 1 digit (total 4)

Done! (When the number becomes 0 your loop should end.)

```
In [8]: num=int(input("Enter a number: "))
        digit_count=0
        for i in range(1,num+1):
            num=num//10
            digit_count+=1
            if num==0:
                break
        print(digit_count)
```

```
Enter a number: 564763
6
```

Task 14

Write a Python program that takes a number from the user and prints the divisors of that number and then print how many divisors were there. [The input number has to be an INTEGER]

==

Example01:

Input:\ 6

Output:\ 1, 2, 3, 6 \ Total 4 divisors.

==

Example02:

Input:\ 121

Output:\ 1, 11, 121 \ Total 3 divisors.

In [10]:

```python
x=int(input("Enter a number: "))
divisor_count=0
for i in range(1,x+1):
    if x%i==0:
        divisor_count+=1
        if i!=x:
            print(i, end=",")
        else:
            print(i, end="")
print()
print("Total",divisor_count,"divisors")
```

```
Enter a number: 121
1,11,121
Total 3 divisors
```

Task 15

Write a Python program that takes a number and tells if it is a perfect number or not. [The input number has to be an INTEGER]

Perfect Number: An integer number is said to be a perfect number if its factors, including 1 but not the number itself, sum to the number.

==

Example01:

Input:\ 6

Output:\ 6 is a perfect number

Explanation:\ 6 have 4 divisors: 1, 2, 3, and 6. \ If we add all factors except itself, 6 = 1 + 2 + 3.

==

Example02:

Input:\ 28

Output:\ 28 is a perfect number

Explanation:\ 28 have 6 divisors: 1, 2, 4, 7, 14, and 28. \ If we add all factors except itself, 28 = 1 + 2 + 4 + 7 + 14.

==

Example03:

Input:\ 33

Output:\ 33 is not a perfect number

Explanation:\ 33 have 3 divisors: 1, 3, 11, and 33. \ If we add all factors except itself, 15 = 1 + 3 + 11.

In [11]:

```
x=int(input("Enter a number: "))
sum_divisor=0
for i in range(1,x+1):
    if x%i==0:
        sum_divisor+=i
if sum_divisor-x==x:
    print(x,"is a parfect number")
else:
    print(x,"is not a parfect number")
```

```
Enter a number: 33
33 is not a parfect number
```

Task 16

Write a Python program that asks the user for one number and tells if it is a prime number or not. [The input number has to be an INTEGER]

Prime Number: If a number has only two divisors, (1 and itself), then it is prime. If it is divisible by more numbers, then it is not a prime.

===

Hint: use the divisor count from task 16.

===

Example01:

Input:\ 11

Output:\ 11 is a prime number

Explanation:\ 11 has only 2 divisors: 1, and 11.

===

Example02:

Input:\ 6

Output:\ 6 is not a prime number

Explanation:\ 6 have 4 divisors: 1, 2, 3 and 6.

In [12]:

```python
x=int(input("Enter a number: "))
divisor_count=0
for i in range(1,x+1):
    if x%i==0:
        divisor_count+=1
if divisor_count==2:
    print(x,"is a prime number")
else:
    print(x,"is not a prime number")
```

```
Enter a number: 11
11 is a prime number
```

Task 17

Write a Python program that asks the user for a quantity then takes that many numbers and prints the maximum, minimum and average of those numbers.

[CANNOT USE MAX,MIN BUILT-IN FUNCTIONS][FOR THIS TASK YOU DO NOT NEED TO USE LISTS]

==

Example01: If the user enters 5 as an input for quantity. Then enters 10, 4, -1, -100, and 1. \ Your program output should be: “Maximum 10”, “Minimum -100”, “Average is -17.2”

Input:\ 5 10 4 -1 -100 1.

Output:\ Maximum 10\ Minimum -100\ Average is -17.2

Explanation:\ Average calculation: (10+4 + (-1) + (-100) + 1)/5 = -86/5=-17.2

In [4]:

```python
loop=int(input("loop: "))
if loop ==1:
    num=int(input("enter a number"))
    count=1
    max_num=num
    min_num=num
    s=num
    while count<loop :
        num=int(input("enter a number"))
        if num>max_num:
            max_num=num
        if num<min_num:
            min_num=num
        s+=num
        count+=1
print("avarage: ",s/loop)
print("maximum: ",max_num)
print("minimum: " ,min_num)
```

```
loop: 5
avarage:  -17.2
maximum:  10
minimum:  -100
```

Task 18

Write a python program that prints a square of size N using + where N will be given as input.

==

Sample Input 5

Sample Output

```
+++++
+++++
+++++
+++++
+++++
```

==

Sample Input 3

Sample Output

```
+++
+++
+++
```

In [13]:

```
N=int(input("Enter numbners of rows: "))

for i in range (N):

    print("+"*N)
```

```
Enter numbners of rows: 3
+++
+++
+++
```

Task 19

Write a python program that prints a rectangle of size M (height/line numbers) and N(length/column numbers) using incrementing numbers where M,N will be given as input.

====================================

Sample Input 4,6

Sample Output
123456
123456
123456
123456

====================================

Sample Input 3,2

Sample Output
12
12
12

In [15]:

```
a,b=map(int,input().split(','))
for i in range(1,a+1):
    for j in range(1,b+1):
        print(j,end="")
    print("")
```

```
4,6
123456
123456
123456
123456
```

Task 20

Write a python program that prints a right angled triangle of height N using incrementing numbers where N will be given as input.

====================================

Sample Input 4

Sample Output
1
12
123
1234

====================================

Sample Input 5

Sample Output
1
12
123
1234
12345

In [16]:

```
n=int(input("Enter n: "))
for i in range(1,n+1):
    for j in range(1,i+1):
        print(j,end="")
    print("")
```

```
Enter n: 4
1
12
123
1234
```

Optional Tasks (21 -26) [Ungraded]

Task 21

Write a python program that prints all the fibonacci number from 0 to N where N will be given.

A Fibonacci number is a number which is the summation of its previous two fibonacci number.

First two fibonacci number are 0 and 1. So the 3rd Fib will be 0+1=1, 4th Fib will be 1+1=2, 5th Fib will be 1+2=3 and so on.

================================

Sample Input
10

Sample Output
0 1 1 2 3 5 8

================================

Sample Input
15

Sample Output
0 1 1 2 3 5 8 13

In [19]:

```python
a=int(input("Enter the N: "))
f=0
s=1
if a<=0:
    print("The requested series is",f)
else:
    print(f,s,end=" ")
    for x in range(2,a):
        next=f+s
        if next>a:
            break
        print(next,end=" ")
        f=s
        s=next
```

```
Enter the N: 15
0 1 1 2 3 5 8 13
```

Task 22

Write a python program that converts a Decimal Integer number to a Boolean Number.

A decimal can be converted to a binary number by keeping track of the remainders after each division of that number by 2.

================================

For example, to convert 10 to a binary number, we can follow the following approach

10/2 = 5 (Remainder 0)
5/2 = 2 (Remainder 1)
2/2 = 1 (Remainder 0)
1/2 = 0 (Remainder 1)

Take the remainders from bottom to top, which is, 1010. Binary of 10 is 1010.

Sample Input
10

Sample Output
1010

================================

For example, to convert 13 to a binary number, we can follow the following approach

13/2 = 6 (Remainder 1)
6/2 = 3 (Remainder 0)
3/2 = 1 (Remainder 1)
1/2 = 0 (Remainder 1)

Take the remainders from bottom to top, which is, 1101. Binary of 13 is 1101.

Sample Input
13

Sample Output
1101

In [20]:

```
def dec_to_bin(num):
    if num > 1:
        dec_to_bin(num // 2)
    print(num % 2, end = '')
number = int(input("Enter the decimal number: "))
dec_to_bin(number)
```

```
Enter the decimal number: 10
1010
```

Task 23

Write a python program that converts a Binary number to a Decimal Integer Number.

A binary number can be converted to its corresponding decimal number by multiplying each binary digit with a power of 2 where the power denotes the position of the binary digit

==

For example, to convert 1010, we can do this -

2^3 x 1 + 2^2 x 0 + 2^1 x 1 + 2^0 x 0 = 10

Sample Input
1010

Sample Output
10

==

For example, to convert 1101, we can do this -

2^3 x 1 + 2^2 x 1 + 2^1 x 0 + 2^0 x 1 = 13

Sample Input
1101

Sample Output
13

In [21]:

```
n = list(input("Enter Binary : "))
value = 0
for i in range(len(n)):
    digit = n.pop()
    if digit == '1':
        value = value + pow(2, i)
print(value)
```

```
Enter Binary : 1101
13
```

Task 24

Take five numbers from the user and find the minimum and the average of **only the even** numbers entered by the user. [If the user enters odd numbers ignore them]

[CANNOT USE MAX,MIN BUILT-IN FUNCTIONS] [FOR THIS TASK YOU DO NOT NEED TO USE LISTS]

Assume, the first input is always an even number.

===

Example01: If the user enters 10, 4, -1, -100, and 1. \ Output: “Minimum -100”, “Average is 28.66667”

Input:\ 10

4

-1

-100

1

Output:\ Minimum -100\ Average is 28.66667

Explanation:\ Average calculation: (10+4 + (-100))/3 = -86/3=-28.66667

===

Example02: If the user enters 2, 10, 1, 21, and 3. \ Output: Minimum 2”, “Average is 6.0”

Input:\ 2

10

1

21

3

Output:\ Minimum 2\ Average is 6.0

Explanation:\ Average calculation: (2+10)/2 = 12/2= 6

In [28]:

```
loop=int(input("loop: "))
count=0
n_new=0
n_old=0
temp=0
sum_new=0
for i in range(5):
    n=int(input("Enter {}th number: ".format(i+1)))
    if abs(n%2)==0:
        sum_new+=n
        count+=1
        n_new=n
        if (i!=0 or i==1) and n_old<n_new:
            temp=n_old
        elif i!=0 or i==1:
            temp=n_new
        elif temp>n_new:
            temp=n_new
        n_old=n_new
print(sum_new/count)
print(temp)
```

```
loop: 5
Enter 1th number: 10
Enter 2th number: 4
Enter 3th number: -1
Enter 4th number: -100
Enter 5th number: 1
-28.666666666666668
-100
```

Task 25

Write a Python code for the following:

- Ask the user to enter a number, N (Total number of inputs to be taken). Assume that the user will never enter the first number as zero.
- Take N number of inputs
- Prints out the product of all the numbers read

For example, if the first input is 4, then the program has to read in four numbers from the user and print the product of these four numbers.

===

Example01:

Input:\ 5

10

6

4

2

1

Output:\ 480

Explanation:\ 5 is the total number of inputs taken. Then, the calculation should be 10 X 6 X 4 X 2 X 1 = 480

In [30]:

```python
N=int(input("Enter N: "))
li=[]
for i in range (N):
    li.append(int(input()))
mul=1
for i in li:
    mul*=i
print(mul)
```

```
Enter N: 5
10
6
4
2
1
480
```

Task 26

Ask the user for a range (a starting number and an ending number). **Count** how many numbers are prime numbers and how many numbers are perfect numbers between that range and **print** those numbers.

==

Hint (1): Declare two strings/lists to store the prime and perfect numbers. Inside the iteration store the values of the prime and perfect numbers in the pre- declared variables.

Hint (2):

ans = "Prime numbers: "\ prime_value = 11 ans = ans + str(prime_value) + " "\ print(ans)

ans = ans + str(13) + " "\ print(ans)

Output:

Prime numbers: 11 \ Prime numbers: 11 13

==

Example:between 2 and 6, there are 3 prime numbers (2, 3, 5) and 1 perfect number (6).

Input:\ 2

6

Output:\ Between 2 and 6,\ Found 3 prime numbers \ Found 1 perfect number\ Prime numbers: 2 3 5 \ Perfect numbers: 6

In [24]:

```
a=int(input('enter the number '))
b=int(input('enter the number '))
perfect_count=0
prime_count=0
prime=''
perfect=''
def find_perfect(n):
    sum=1
    for i in range(2,n//2+1):
        if(n%i==0):
            sum=sum+i
    if(sum==n):
        return True,n
    else:
        return False,n
def find_prime(num):
    if num > 1:
        for i in range(2, num):
            if (num % i) == 0:
                return False, num
                break
        else:
            return True,num
    else:
        return False,num
for i in range(a,b+1):
    if find_perfect(i)[0]:
        perfect_count+=1
        perfect=perfect+ str(i)+' '
    if find_prime(i)[0]:
        prime_count += 1
        prime=prime+ str(i)+' '
print(f"Between {a} and {6}")
print(f"Found {prime_count} prime numbers")
print(f"Found {perfect_count} perfect numbers")
print(f"Prime nimbers {prime}")
print(f"Prefect Number: {perfect} ")
```

```
enter the number 2
enter the number 6
Between 2 and 6
Found 3 prime numbers
Found 1 perfect numbers
Prime nimbers 2 3 5
Prefect Number: 6
```

In [ ]: