**DSA Assignment-1**

**Abhinav Masih**

**BTAI-301**

**Github -**

1. Find the largest of two numbers.

**Algorithm-**

1. Start.
2. Initialize two variables.
3. Input 2 values and store in variables.
4. Compare both variables and print the greater result.
5. End.

**Code-**

**c**

#include <stdio.h>

int main(){

    int a,b;

    printf("Enter First num:");

    scanf("%d",&a);

    printf("Enter second num:");

    scanf("%d",&b);

    if (a > b) {

        printf("%d is greater",a);

    }

    else{

        printf( "%d is greater",b);

    }

    return 0;

}

**Output-**

****

**Python**

a = int(input('Enter first #:'))

b = int(input('Enter second #:'))

if a == b:

    print('Values are equal')

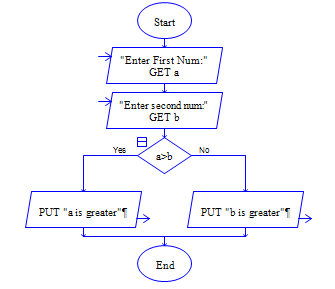
else:

    print(a if a > b else b)

**Output-**



**Flowchart-**

****

2. Check whether a number is even or odd.

**Algorithm-**

1. Start.
2. Initialize a variable .
3. Input value and store in variable.
4. Using conditional statements check if the variable is perfectly divisible by 2, if so print “Even”, otherwise “odd”.
5. End.

**Code-**

**C**

#include <stdio.h>

int main(){

    int a;

    printf("Enter Num:");

    scanf("%d",&a);

    if (a % 2 == 0){

        printf("%d is even",a);

    }

    else{

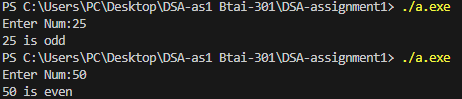
        printf("%d is odd",a);

    }

    return 0;

}

**Output-**

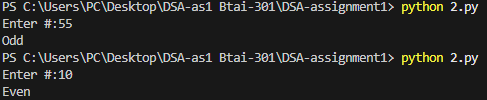
****

**Python-**

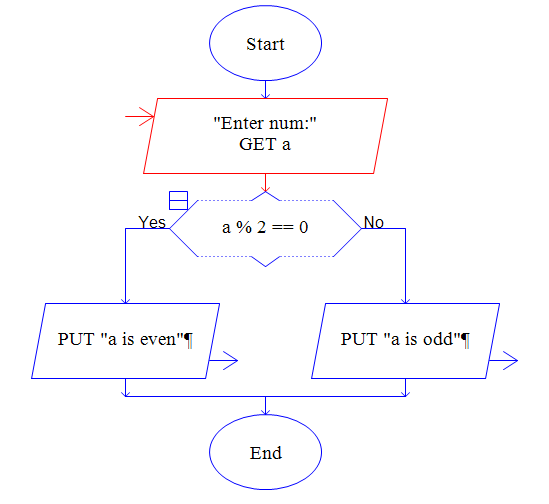
num = int(input('Enter #:'))

print("Even" if num % 2 ==0 else "Odd")

**Output**

****

**Flowchart-**



3. Find the factorial of a number

**Algorithm-**

1. Start.
2. Initialize two variables.
3. Input value and store in one variable.
4. Use a loop with exit condition a <= 0, multiply other variable with input variable and decrement the input variable.
5. Print the result.
6. End.

**Code-**

**C-**

#include <stdio.h>

int main(){

    int a,fact=1;

    printf("Enter num:");

    scanf("%d",&a);

    for (; a > 0; a--) {

        fact = fact \* a;

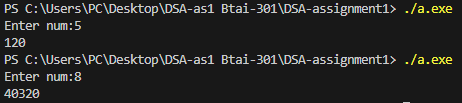
    }

    printf("%d",fact);

    return 0;

}

**Output-**

****

**Python-**

num = int(input('Enter #:'))

fact = 1

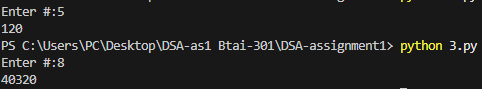
for i in range(1,num+1):

    fact = fact \* i

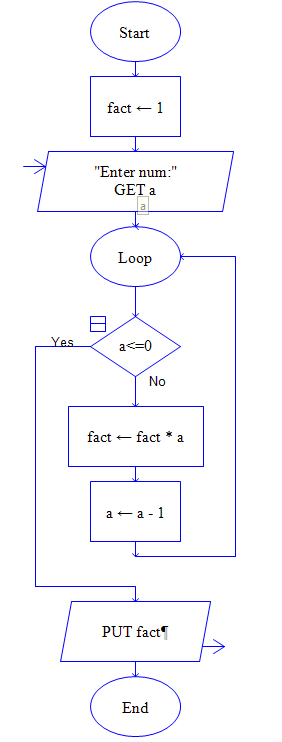
    i += 1

print(fact)

**Output-**



**Flowchart-**



4. Calculate the sum of first N natural Numbers

**Algorithm-**

1. Start.
2. Initialize two variables.
3. Input value and store in a variable.
4. Use for loop with exit condition input variable <= 0, add input variable to other variable and decrement input variable.
5. Print result
6. End

**CODE:**

**C –**

// sum of First N numbers

#include <stdio.h>

int main(){

    int a,sum=0;

    printf("Enter num:");

    scanf("%d",&a);

    for (;a>0;a--){

        sum = sum + a;

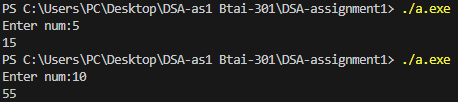
    }

    printf("%d", sum);

    return 0;

}

**Output-**



**Python –**

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

sum=0

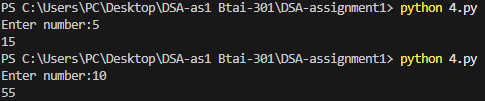
for i in range(1,num+1):

    sum += i

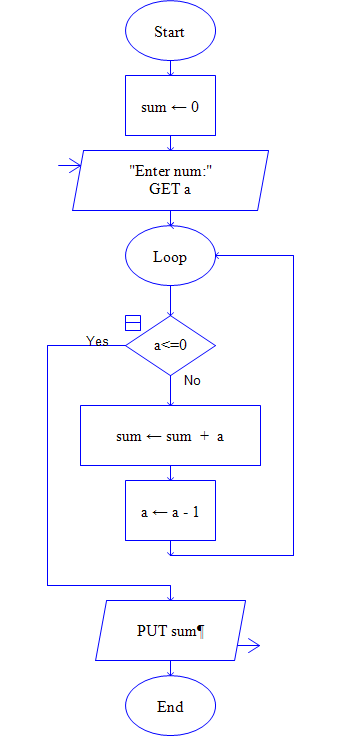
    i+=1

print(sum)

**Output-**

****

**Flowchart –**

****

5. Check whether a number is prime or not.

**Algorithm-**

1. Start.
2. Initialize two variables, one being factors = 1
3. Input value and store in a variable
4. Use a loop with exit condition loop variable >= input variable and using conditional statements check if input variable is perfectly divisible by loop variable, if so increment factors variable, also increment loop variable.
5. Using conditional statements check if factors are > 2, if so the number is not prime, otherwise the number is prime.
6. Print result
7. End

**CODE:**

**C –**

#include <stdio.h>

int main(){

    int a, factors = 1;

    printf("Enter #:");

    scanf("%d",&a);

    for(int i=1;i<a;i++){

        if (a % i == 0) {

            factors++;

        }

    }

    if (factors > 2){

        printf("Is not prime");

    }

    else{

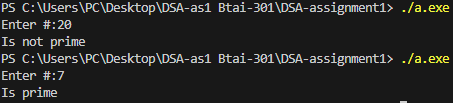
        printf("Is prime");

    }

    return 0;

}

**Output-**

****

**Python-**

num  = int(input("Enter #:"))

factors = 0

if num ==0 or num ==1:

    print("Neither prime nor composite")

else:

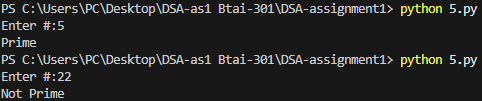
    for i in range(1,num+1):

        if num % i == 0 :

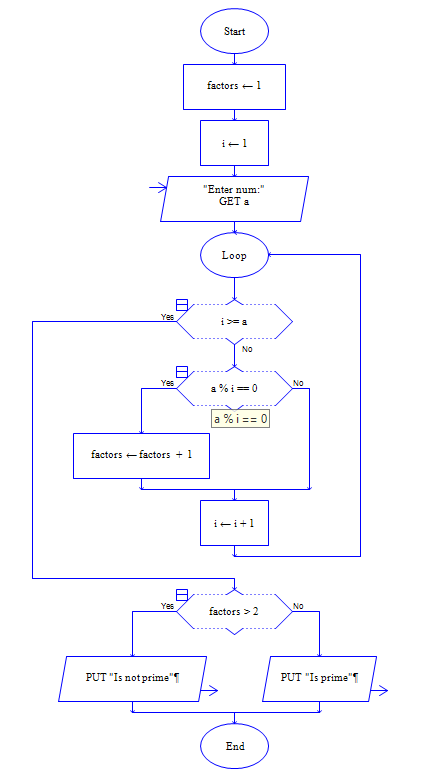
            factors += 1

    print("Prime" if factors == 2 else "Not Prime")

**Output-**

****

**Flowchart-**

****

6. Reverse a number

**Algorithm-**

1. Start.
2. Initialize two variables, one being ld = 1
3. Input value and store in variable.
4. Use loop with exit condition input variable <= 0, ld = input variable % 10 and divide input variable by 10. Print ld
5. End

**Code:**

**C –**

#include <stdio.h>

int main(){

    int a,ld=1;

    printf("Enter number:");

    scanf("%d",&a);

    while( a > 0){

        ld = a % 10;

        a = a / 10;

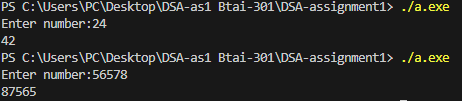
        printf("%d",ld);

    }

    return 0;

}

**Output -**

****

**Python -**

num = int(input("Enter #:"))

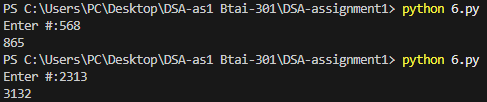
while num > 0:

    ld = num % 10

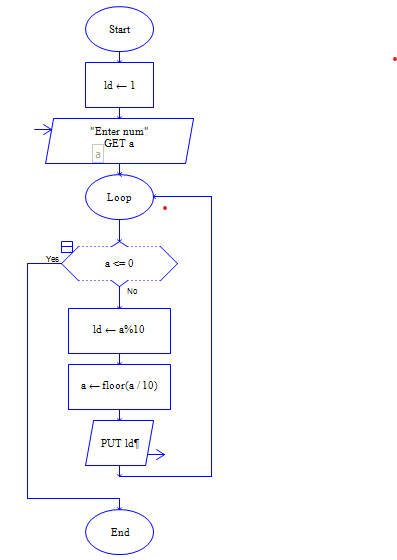
    num = num // 10

    print(ld,end="")

**Output –**



**Flowchart –**

****

7. Generate Fibonacci series up to N terms.

**Algorithm**

1. Initialize 4 variable, one being ld=1, another ld2=0, temp=0(ld, ld2 represent the starting of Fibonacci series 0,1).
2. Input value and store in variable
3. Using conditional statement check is input value is 0 or 1, if so print input value
4. Otherwise print ld2 and ld to start the series
5. Use a for loop with exit condition I >=n and loop variable set to 2, assign temp to ld, ld to ld+ld2 and ld2 to temp,print ld at each iteration and increment loop vairble
6. End

**Code:**

**C –**

#include <stdio.h>

int main(){

    int ld=1,ld2=0,n,temp=0;

    printf("Enter number:");

    scanf("%d",&n);

    if (n ==0 || n == 1){

        printf("%d",n);

    }

    else{

        printf("%d ",ld2);

        printf("%d ",ld);

        for(int i = 2;i < n ; i++){

            temp = ld;

            ld = ld+ld2;

            ld2 = temp;

            printf("%d ",ld);

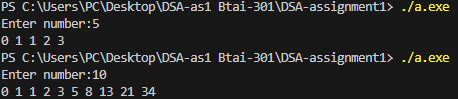
        }

    }

    return 0;

}

**Output –**



**Python –**

num = int(input("Enter #:"))

ld = 1

ld2 = 0

if num == 0 or num == 1:

    print(num)

else:

    print(ld2,end=" ")

    print(ld,end=" ")

    for i in range(2,num):

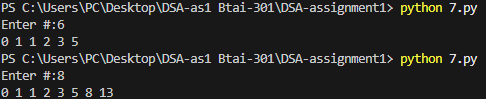
        temp = ld

        ld += ld2

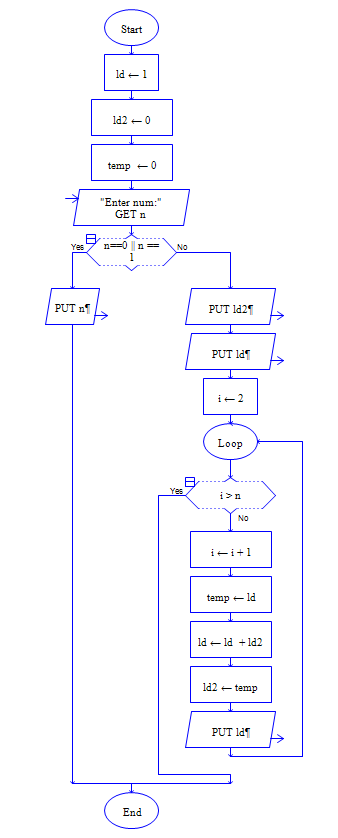
        ld2 = temp

        print(ld,end=" ")

**Output -**



**Flowchart –**

****

8. check whether a number is palindrome

**Algorithm**

1. Initialize 4 variables
2. Input value and store in variable.
3. Assign input variable to temp variable
4. Use loop with exit condition temp var == 0 get last digit using % user variable and store in a variable, divide temp by 10 and in a result variable assign result \* 10 + last digit.
5. Using conditional statements check if input variable is same as result variable, if so print the input variable is palindrome, otherwise not palindrome
6. Print result
7. End

**CODE:**

**C –**

#include <stdio.h>

int main(){

    int a, ld, rev=0, temp=0;

    printf("Enter number:");

    scanf("%d",&a);

    temp = a;

    while( temp != 0){

        ld = temp % 10;

        temp = temp / 10;

        rev  = rev \* 10 + ld;

    }

    if (a == rev){

        printf("%d Is Palindrome",rev);

    }

    else{

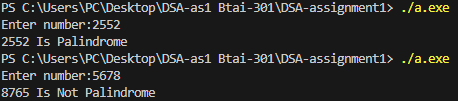
        printf("%d Is Not Palindrome",rev);

    }

    return 0;

}

**Output:**



**Python –**

num = int(input("Enter #:"))

temp = abs(num)

reverse = 0

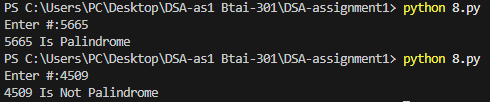
while temp !=0 :

    reverse = (reverse \* 10) + (temp % 10)

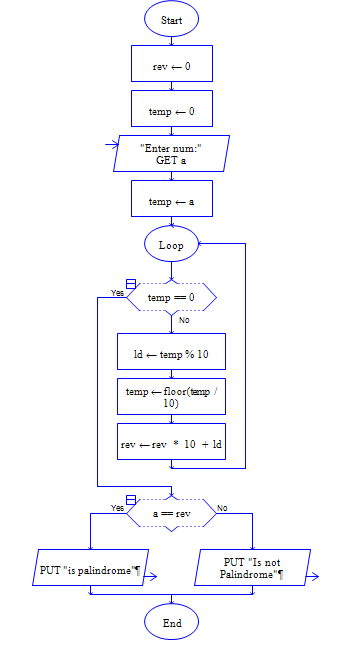
    temp = temp // 10

print(f"{num} Is Palindrome" if reverse == num else f"{num} Is Not Palindrome")

**Output –**

****

**Flowchart –**

****

9. Convert Celsius to Fahrenheit

**Algorithm:**

1. Initialize two variables
2. Input Celsius value
3. Convert Celsius to Fahrenheit using formula (9/5) \* Celsius + 35
4. Print result
5. End

**Code:**

**C –**

#include <stdio.h>

int main(){

    float cel,far=1;

    printf("Enter Value:");

    scanf("%f",& cel);

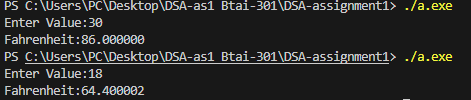
    far = ((9.0/5.0) \* cel) + 32;

    printf("Fahrenheit:%f",far);

    return 0;

}

**Output-**

****

**Python –**

cel = input("Enter value:")

try:

    cel = float(cel)

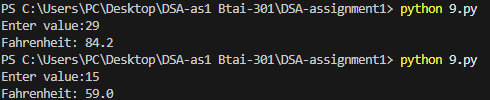
    far = (9/5 \* cel) + 32

    print("Fahrenheit:", far)

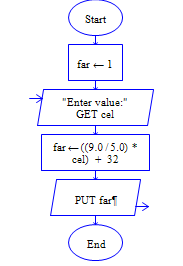
except ValueError:

    print("Invalid input!")

**Output –**

****

**Flowchart-**

****

10. Create a simple calculator.

**Algorithm:**

1. Start
2. Initialize 3 variables, two for values and one for operator.
3. Input values and operator and store in variables
4. Using conditional statements check for matching operator from operator list [+, -, \*, %], if match perform operation and print result otherwise print invalid operator
5. End

**Code:**

#include <stdio.h>

*int* main(){

*float* num1,num2;

*char* operator;

    printf("Enter first number:");

    scanf("%f", &num1);

    printf("Enter second number:");

    scanf("%f", &num2);

    printf("Enter Operator(+ - \* /):");

    scanf(" %c",&operator);

    if (operator == '+'){

        printf("Result: %f",num1 + num2 );

    }

    else if (operator == '-'){

        printf("Result: %f",num1 - num2 );

    }

    else if (operator == '\*'){

        printf("Result: %f",num1 \* num2 );

    }

    else if (operator == '/'){

        if (num2 != 0)

            printf("Result: %f\n", num1 / num2);

        else

            printf("Error: Division by zero!\n");

    }

    else{

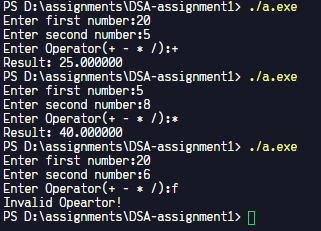
        printf("Invalid Opeartor!" );

    }

    return 0;

}

**Output:**

****

**Python –**

num1 = input("Enter First Num:")

num2 = input("Enter Second Num:")

operator = input("Enter Operator(+ - \* /):")

try:

    num1 = *float*(num1)

    num2 = *float*(num2)

    if operator == '+':

        print('Result: ',num1 + num2)

    elif operator == '-':

        print('Result: ',num1 - num2)

    elif operator == '\*':

        print('Result: ',num1 \* num2)

    elif operator == '/':

        print('Result: ',num1 / num2)

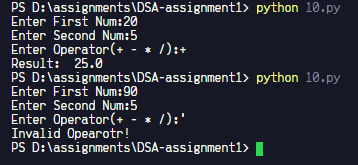
    else:

        print('Invalid Opearotr!')

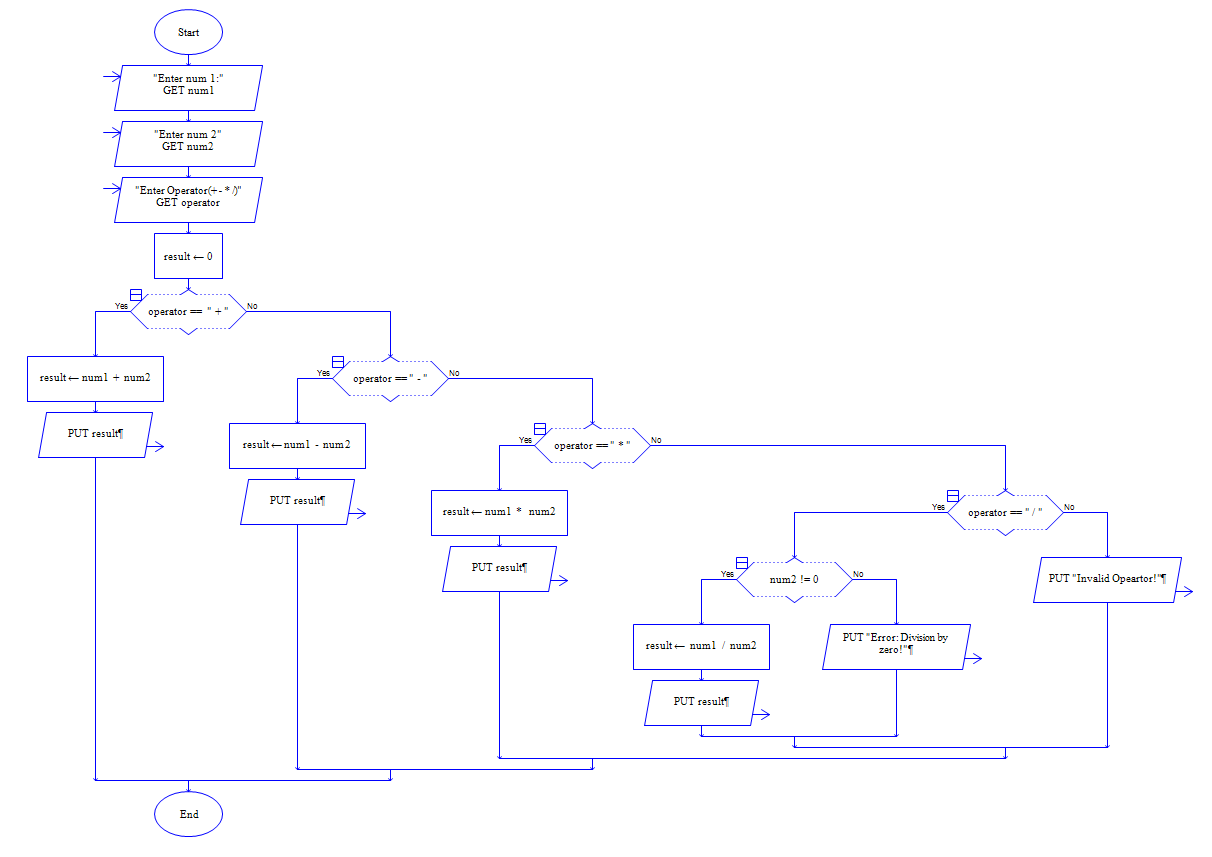
except ValueError:

    print("Invalid input!")

**Output:**

****

**Flowchart:**

****

11. Find GCD of two numbers

**Algorithm:**

1. Start
2. Initialize 3 variables
3. Input values and store in variables
4. Find upper bound for loop, select the minimum value from two input and store in a variable.
5. Use loop with exit condition loop variable == 0 & loop variable’s value set to upper bound, check if both the 2 input values are perfectly divisible by I, if so value of I represents GCD for the 2 input variables, print the result and break loop, otherwise decrement I.
6. End

**Code:**

**C –**

#include <stdio.h>

*int* main(){

*int* num1,num2;

    printf("Enter num1:");

    scanf("%d",&num1);

    printf("Enter num2:");

    scanf("%d",&num2);

*int* ub = (num1 < num2) ? num1:num2;

    for(*int* i = ub;i > 0 ; i--){

        if(( num1 % i == 0  ) && ( num2 % i == 0)){

            printf("GCD of %d & %d: %d",num1,num2,i);

            break;

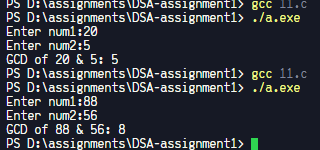
        }

    }

    return 0;

}

**Output:**

****

**Python –**

num1 = *int*(input("Enter num1:"))

num2 = *int*(input("Enter num2:"))

ub = num1 if num1 > num2 else num2

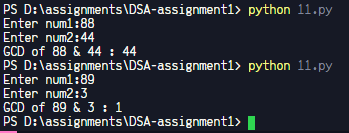
for i in range(ub,0,-1):

    if num1 % i == 0 and num2 % i == 0:

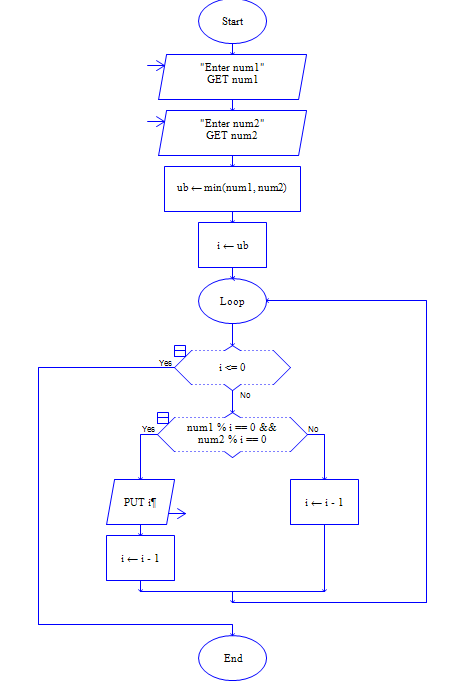
        print(*f*"GCD of {num1} & {num2} : {i}")

        break

**Output:**

****

**Flowchart:**

****

12. Count the number of digits in a number

**Algorithm:**

1. Start
2. Initialize 2 variables, one with value 0
3. Input value and store in variable
4. Check if values entered is 0, if so print 1 and end program
5. Use loop with exit condition input variable < 0, divide num by 10 and increment the counter variable
6. Print the result
7. End

**CODE:**

**C –**

#include <stdio.h>

*int* main(){

*int* num, dg=0;

    printf("Enter Num:");

    scanf("%d",&num);

    if (num == 0) {

        dg = 1;

    }

    else{

        while (num > 0){

            num = num / 10;

            dg++;

        }

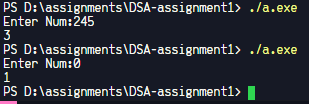
    }

    printf("%d",dg);

    return 0;

}

**Output:**



**Python:**

# count digit

num = *int*(input("Enter Value:"))

dg = 0

if num == 0:

    dg = 1

else:

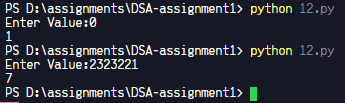
    while num > 0:

            num = num // 10

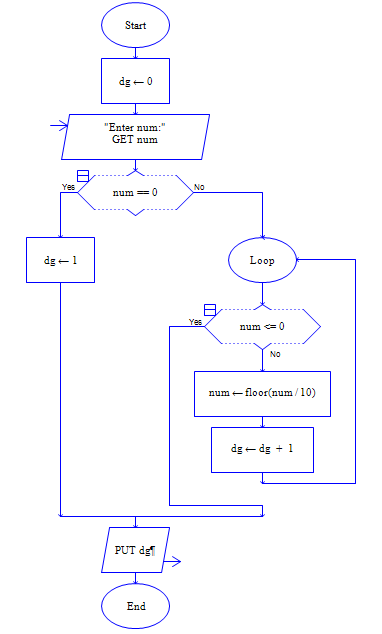
            dg += 1

print(dg)

**Output:**



**Flowchart:**



13. check whether a year is a leap year.

**Algorithm:**

1. Start
2. Initialize a variable
3. Input Year and store in variable
4. Check if year is perfectly divisible by 4 and not perfectly divisible by 100 or check if year is perfectly divisible by 400. If any of the condition return true then the input year is leap year
5. Print Result
6. End

**Code:**

**C –**

#include <stdio.h>

*int* main(){

*int* year;

    printf("Enter Year:");

    scanf("%d",&year);

    if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {

        printf("%d is a leap year\n", year);

    } else {

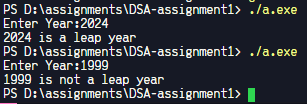
        printf("%d is not a leap year\n", year);

    }

    return 0;

}

**Output:**

****

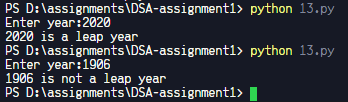
**Python**

year = *int*(input("Enter year:"))

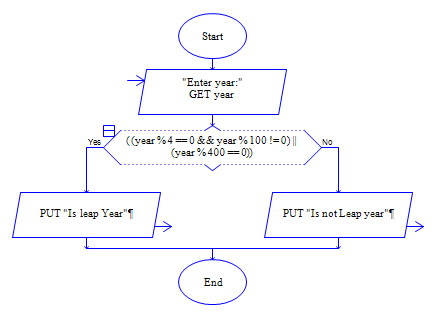
isYear = *lambda* *x*: ((*x* % 4 == 0 and *x* % 100 != 0) or (*x* % 400 == 0))

print( *f*"{year} is a leap year" if isYear(year) else *f*"{year} is not a leap year")

**Output –**

****

**Flowchart –**

****

14. calculate average of N numbers

**Algorithm:**

1. Start
2. Initialize 3 variables with 2 variable with value 0
3. Input number of elements and store in a variable
4. Check if number of elements is not a valid number(less than 0 or 0), if so print “Invalid input” and End program
5. Create an array of size number of elements
6. Using loop with exit condition loop variable < number of elements, take inputs into the array.
7. Using loop with exit condition loop variable < number of elements Add each element to total variable
8. Compute mean = total / n
9. Print the value of mean
10. End

**Code:**

**C –**

#include <stdio.h>

*int* main(){

*int* n,total=0,mean=0;

    printf("Enter number of elements:");

    if (scanf("%d", &n) != 1 || n <= 0) { // if the user enters a valid integer, scanf returns 1

        printf("Invalid input. Please enter a positive integer.\n");

        return 0;

    }

*int* elements[n];

    for (*int* i = 0; i < n; i++){

        printf("Element %d:", i+1);

        scanf("%d", &elements[i]);

    }

    for (*int* i=0; i < n ; i++){

        total += elements[i];

    }

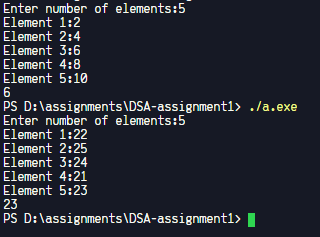
    mean = total/n;

    printf("%d",mean);

    return 0;

}

**Output:**

****

**Python:**

**user\_input = input("Enter elements separated by spaces: ")**

**elements = user\_input.split()**

**try:**

**elements = *list*(map(*lambda* *x*: *float*(*x*), elements))**

**lenght = len(elements)**

**total = sum(elements)**

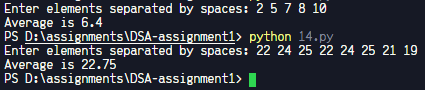
**mean = total/lenght**

**print(*f*"Average is {mean}")**

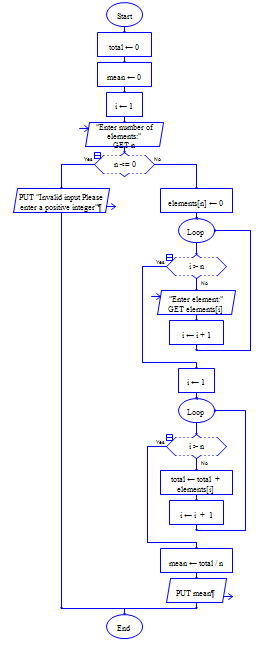
**except ValueError:**

**print("Enter numeric values only!")**

**Output:**

****

**Flowchart –**

****

15. Print multiplication table of a given number

**Algorithm-**

1. Start
2. Initialize a variable
3. Input value and store in variable
4. Using loop with loop variable = 1 and loop variable > 10 as exit condition print loop variable multiplied by input variable
5. End

**Code:**

**C** **–**

#include <stdio.h>

*int* main(){

*int* num;

    printf("Enter value:");

    scanf("%d",&num);

    for (*int* i =1; i < 11; i++){

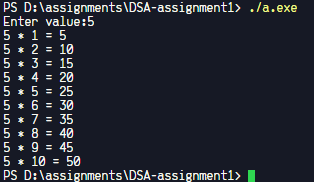
        printf("%d \* %d = %d\n", num, i, num \* i);

    }

    return 0;

}

**Output:**



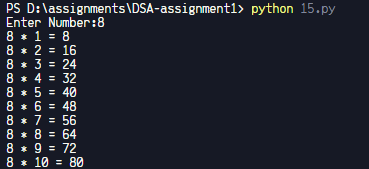
**Python –**

num = *int*(input("Enter Number:"))

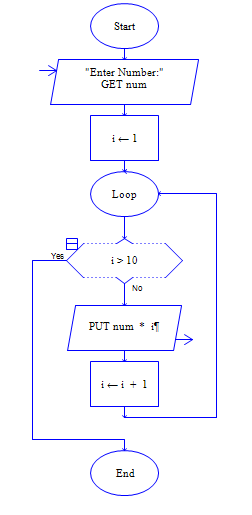
for i in range(1,11):

    print(*f*"{num} \* {i} = {num \* i}")

**Output:**



**Flowchart:**

****