**DSA Assignment-1**

**Abhinav Masih**

**BTAI-301**

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')

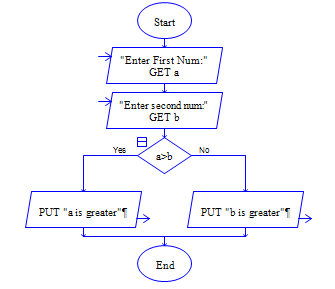
else:

    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{

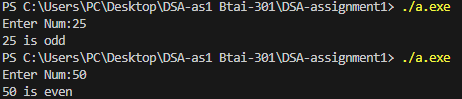
        printf("%d is odd",a);

    }

    return 0;

}

**Output-**

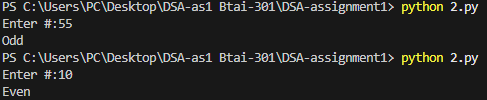
****

**Python-**

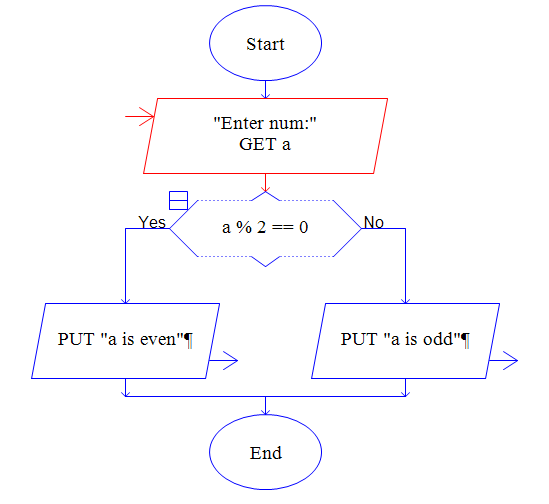
num = int(input('Enter #:'))

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;

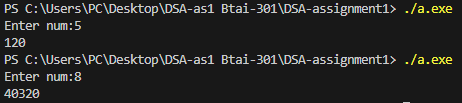
    }

    printf("%d",fact);

    return 0;

}

**Output-**

****

**Python-**

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

fact = 1

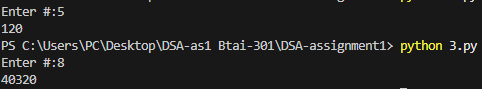
for i in range(1,num+1):

    fact = fact \* i

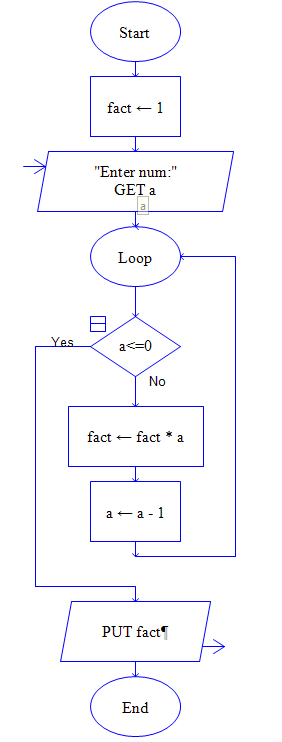
    i += 1

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;

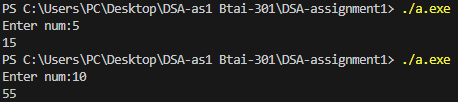
    }

    printf("%d", sum);

    return 0;

}

**Output-**



**Python –**

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

sum=0

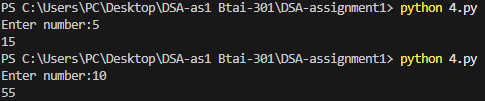
for i in range(1,num+1):

    sum += i

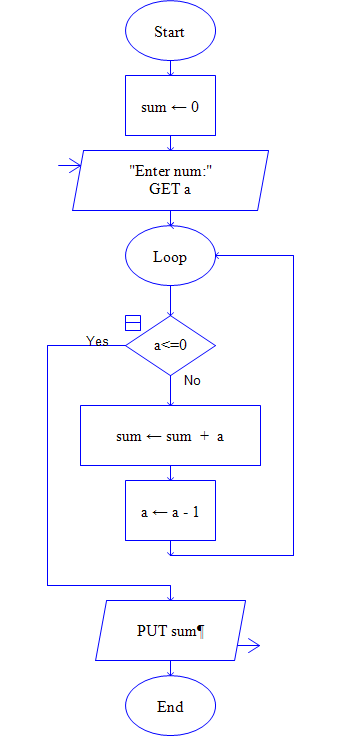
    i+=1

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{

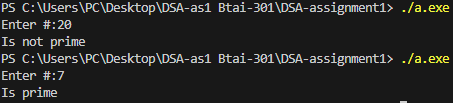
        printf("Is prime");

    }

    return 0;

}

**Output-**

****

**Python-**

num  = int(input("Enter #:"))

factors = 0

if num ==0 or num ==1:

    print("Neither prime nor composite")

else:

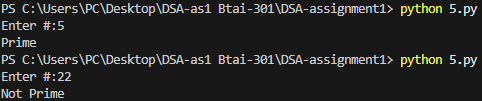
    for i in range(1,num+1):

        if num % i == 0 :

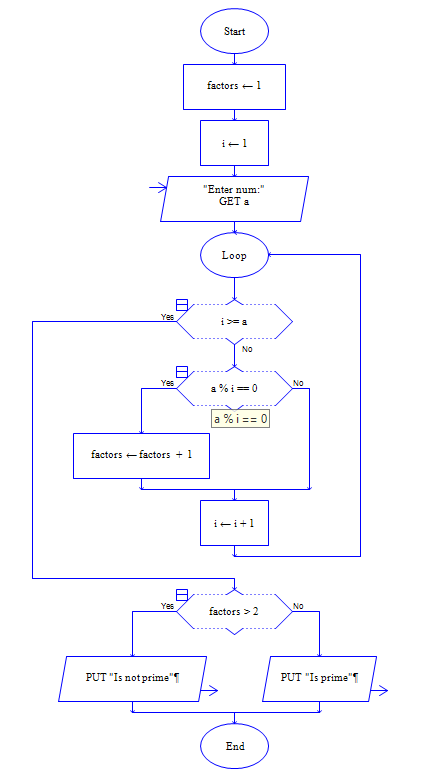
            factors += 1

    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;

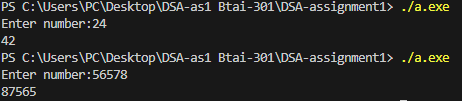
        printf("%d",ld);

    }

    return 0;

}

**Output -**

****

**Python -**

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

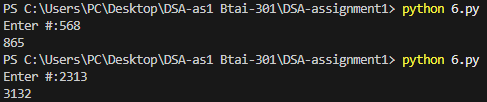
while num > 0:

    ld = num % 10

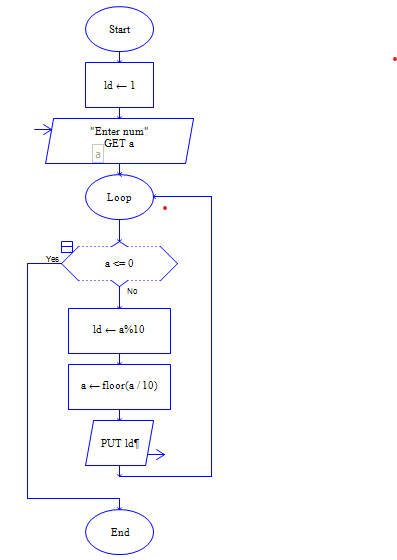
    num = num // 10

    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);

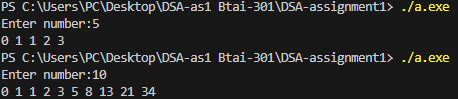
        }

    }

    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):

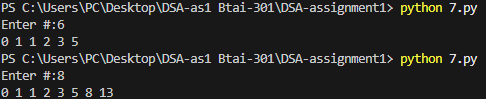
        temp = ld

        ld += ld2

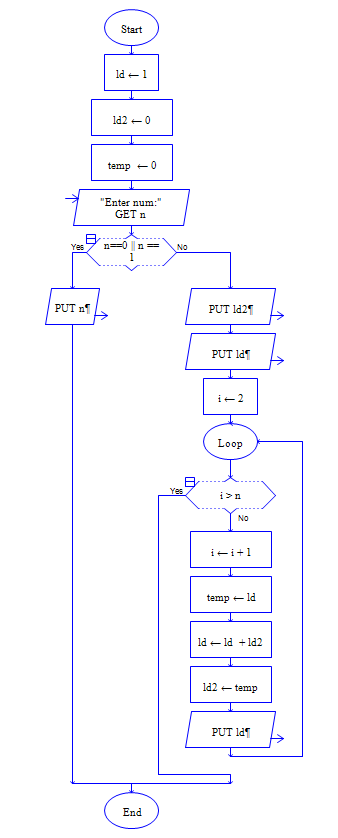
        ld2 = temp

        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{

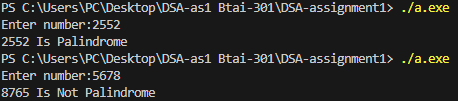
        printf("%d Is Not Palindrome",rev);

    }

    return 0;

}

**Output:**



**Python –**

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

temp = abs(num)

reverse = 0

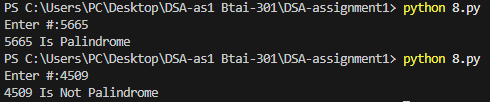
while temp !=0 :

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

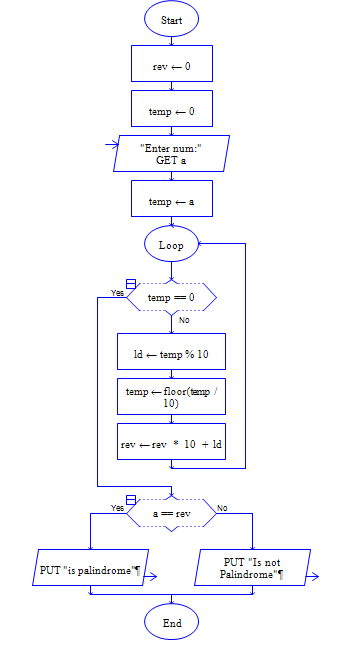
    temp = temp // 10

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);

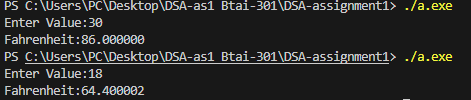
    far = ((9.0/5.0) \* cel) + 32;

    printf("Fahrenheit:%f",far);

    return 0;

}

**Output-**

****

**Python –**

cel = input("Enter value:")

try:

    cel = float(cel)

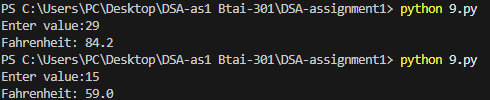
    far = (9/5 \* cel) + 32

    print("Fahrenheit:", far)

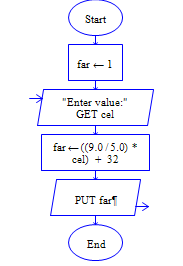
except ValueError:

    print("Invalid input!")

**Output –**

****

**Flowchart-**

****