Assignment -2

# Write a recursive function pseudocode and Calculate the nth Fibonacci number and use Big O notation to analyze its Efficiency.Compare this with an iterative approach and discuss the pros and Cons in terms of space and time complexity.

Pseudocode :

1. Start

2. Initialize i, v1, v2, Fb to 0, 0, and 1, respectively

3. Prompt the user to enter the nth number

4. Call the function FS with arguments v1, v2, and n

5. In the FS function:

If n equals 0, return 1

Calculate Fb as the sum of a1 and a2

6.Print Fb

Update a1 and a2 with the values of a2 and Fb, respectively

Decrement n by 1

Recursively call FS with updated values of a1, a2, and n

7.End Program

Algorithm:

1.start

2.declare i=1,v1,v2,fb

3.Set v1=0,v2=1,fb=0

4.read n

5.print v1,v2

6.use loop until i<=n

Fb=v1+v2

Set v1=v2

Set v2=fb

I=i+1

7.print fb

8.end

Code:

#Include <stdio.h>

Int FS(int,int, int);

Int main()

{

Int i=0,v1=0,v2=1,Fb=0,n; Printf(“\nEnter the nth Number: “); Scanf(“%d”,&n);

Printf(“\nFib Series of %dth Number is\n”,n);

Printf(“%d “,v2);

/\*while(i<n)

  {

Fb = v1+v2;

      V1 = v2;

      V2 = Fb;

      Printf(“%d “,Fb);

      I++;

    }

    \*/

FS(v1,v2,n);

    Printf(“\n\n”);

    Return 0;

}

Int FS(int a1,int a2,int n)

{

Int Fb;

    If (n==0)

{

Return 1;

}

    Fb = a1+a2;

    Printf(“%d “,Fb);

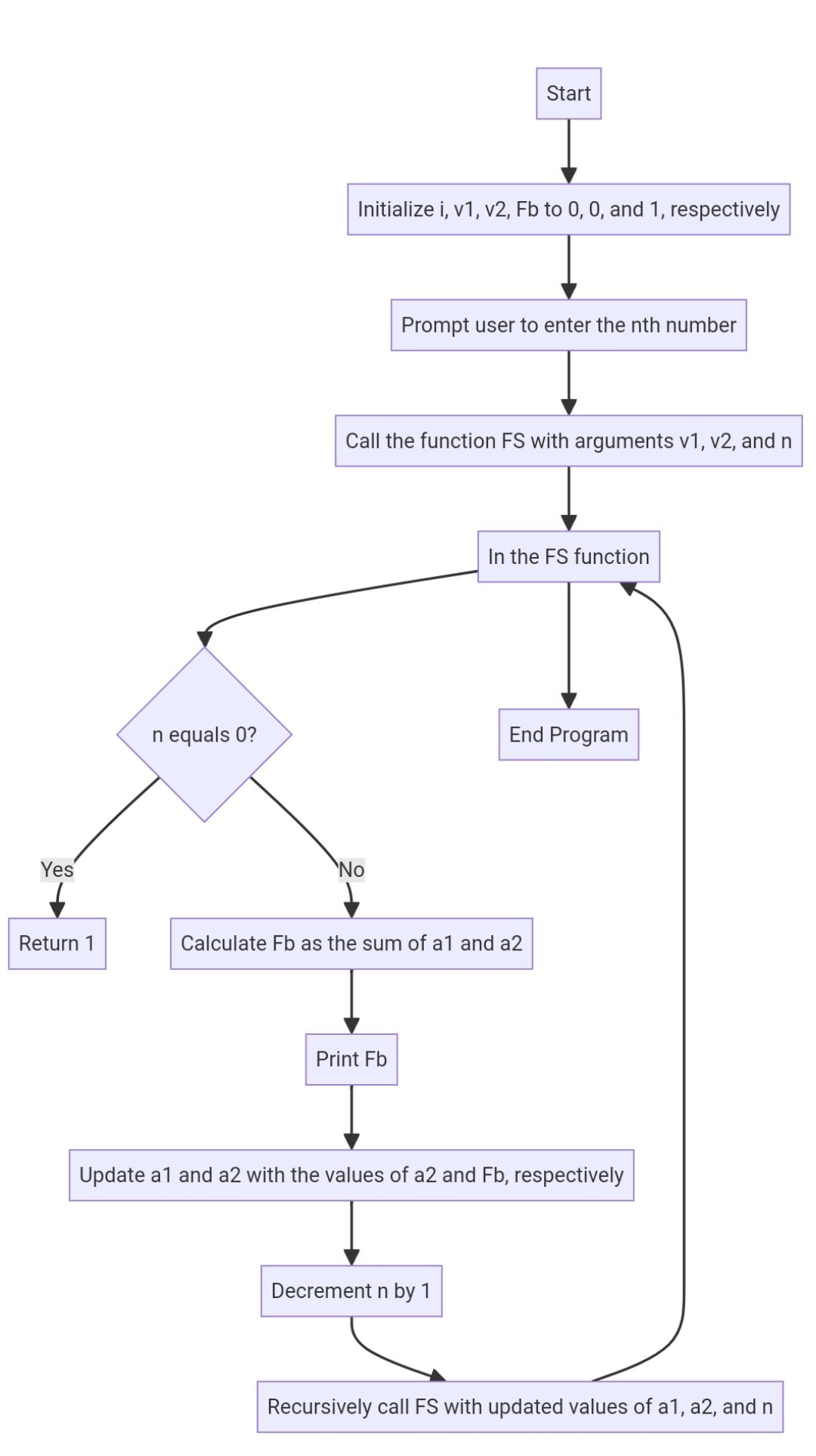
    A1 = a2;

    A2 = Fb;

    N=n-1;

    FS(a1,a2,n);

}



FLOWCHART