Assignment No – 01

Consider the following Fibonacci series and solve the following conditions

fib (n) = fib(0), fib (1), fib (2),……….fib(n)

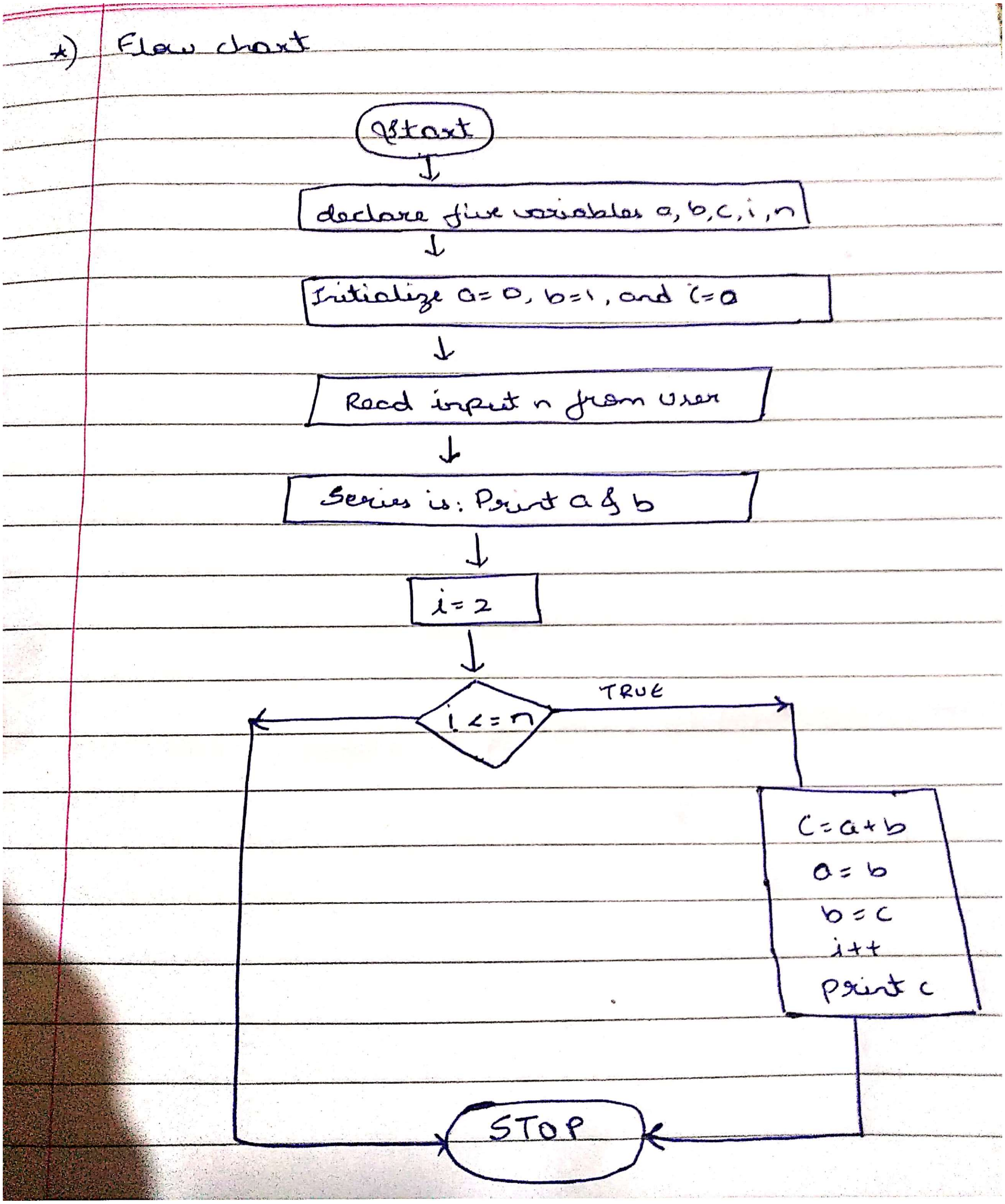
where fib(n) = fib(n-1) + fib(n-2)

a) Draw the Flow chart , Algorithms in pseudo code for solving .

Soln :

```
Step 1: Start

Step 2: Declare variables a,b,c,n,i

Step 3: Initialize variable a=0, b=1, i=2

Step 4: Read n from user

Step 5: Print a and b

Step 6: Repeat until i<=n
            c=a+b
            print c
            a=b, b=c
            i=i+1

Step 7: Stop .
```

Flow chart :


*) Flow chart
Start
declare five variables a, b, c, i, n
Initialize a = 0, b = 1, and i = 0
Read input n from user
Series is: Print a & b
i = 2
i <= n
TRUE
c = a + b
a = b
b = c
i++
print c
STOP

b) Write two types of algorithm (recursive and non recursive) for fib(5) and fib(500) series

Soln : Recursive Algorithm for fib(5)

Step-1 : Start

Step-2 : declare variables a, b, c, i and n

Step-3 : Intialize a=0, b=1, i=2 and n=5

Step-4 : Print a and b

Step-5 : if(i>n) then go to step 12

Step-6 : c = a+b

Step-7 : Print c

Step-8 : a=b

Step-9 : b=c

Step-10 : i=i+1

Step-11 : Go to step-5

Step-12 : Stop.

Recursive Algorithm for fib(500)

Step-1 : Start

Step-2 : declare variables a, b, c, i and n

Step-3 : Intialize a=0, b=1, i=2 and n=500

Step-4 : Print a and b

Step-5 : if(i>n) then go to step 12

Step-6 : c = a+b

Step-7 : Print c

Step-8 : a=b

Step-9 : b=c

Step-10 : i=i+1

Step-11 : Go to step-5

Step-12 : Stop.

Iterative Algorithm for fib(5)

Step 1: Start

Step 2: Declare variables a,b,c,n,i

Step 3: Initialize variable a=0, b=1, i=2 and n=5

Step 4: Print a and b

Step 5: Repeat until i<=n
    c=a+b
    print c
    a=b, b=c
    i=i+1

Step 6: Stop .

Iterative Algorithm for fib(500)

Step 1: Start

Step 2: Declare variables a,b,c,n,i

Step 3: Initialize variable a=0, b=1, i=2 and n=500

Step 4: Print a and b

Step 5: Repeat until i<=n
    c=a+b
    print c
    a=b, b=c
    i=i+1

Step 6: Stop .

c) Find out the Total memory or space required to perform these Fibonacci series computational operations

Soln :

For Iterative method :

Space Required / Total Memory = 4 Bytes * 5 variables

= 20 Bytes

Therefore , Space complexity is O(1) / O(Constant Space).

For Recursion method :

Space Required / Total Memory = 4 Bytes * 5 variables + O(n)

= 20 Bytes + O(n)

= O(n) .

[There will be n recursive calls , so there will be n stacks used. Hence O(n) Space]

Therefore , Space complexity is O(n) .

d) Find out the WORST CASE and BEST CASE scenario from the above identified approaches

Soln :

Recursive Fibonacci Algorithm holds the worst case scenario , has it occupies O(n) space , the total memory consumption depends on the n .

Iterative Fibonacci Algorithm holds the best case scenario , has it occupies O(1) space / O(Constant space) , the total memory consumption doesn't depend on the n .

e) Write a program and compare the actual memory consumed by all the

approaches

Soln : Iterative code :

```
import os

import psutil
def fib1(n):
    a = 0
    b = 1
    if n < 0:
        print("Incorrect input")
    elif n == 0:
        return a
    elif n == 1:
        return b
    else:
        for i in range(2,n):
            c = a + b
            a = b
            b = c
            i  = i + 1
        return b
k = int(input("enter Fibonacci sequence index number: "))
print(fib(k))
process = psutil.Process(os.getpid())
print(process.memory_info().rss)
```

Memory consumed by this approach is

Recursion code :

```
import os
import psutil
```

```
def fib(n):

    If n <0:
        print("incorrect input")
    elif n == 0:
         return 0
    elif n == 1:
         return 1
    else:
         return fib(n-1)+fib(n-2)

k = int(input("enter Fibonacci sequence index number: "))
print(fib(k))
process = psutil.Process(os.getpid())
print(process.memory_info().rss)
```

Memory consumed by this approach is