C12

1.

Algorithm

```
a. Get the number from the user
b. If the number is >= 1 then
    i. For I in 1 .. number loop
        ▪ Factorial := Factorial * I;
c. Else
    i. Display Cannot Compute Factorial
d. Display Computed Factorial to the User
```

Code Listing

GNAT 3.13p (20000509) Copyright 1992-2000 Free Software Foundation, Inc.

Compiling: c:/docume~2/ joeb /desktop/16070/codeso~1/factorial_with_iteration.adb (source file time stamp: 2003-10-02 03:57:26)

```
-----------------------------------------------------
-- Program to find the factorial of a number using
-- iteration.
-- Programmer : Joe B
-- Date Last Modified : October 01, 2003
-----------------------------------------------------


with Ada.Text_Io;
with Ada.Integer_Text_Io;

procedure Factorial_With_Iteration is
   Factorial : Integer := 1; -- initializise factorial to 1
   Num       : Integer;      -- variable used to get input from the user
begin
   -- get the number from the user
   Ada.Text_Io.Put("Please Enter A Number : ");
   Ada.Integer_Text_Io.Get(Num);
   Ada.Text_Io.Skip_Line;

   if Num >=1 then

      -- compute factorial
      for I in 1 .. Num loop
         Factorial:= Factorial * I;
      end loop;

      -- display the computed factorial to the user
      Ada.Text_Io.Put("The Factorial of ");
      Ada.Text_Io.Put(Integer'Image(Num));
      Ada.Text_Io.Put("is : ");
      Ada.Integer_Text_Io.Put(Factorial);

   else
```

```
      Ada.Text_Io.Put("Cannot Compute Factorial");
   end if;

end Factorial_With_Iteration;
```

```
38 lines: No errors
```

2.

The program does not work when you have numbers greater than 12 because the value of the factorial is larger than Integer’Last

3.

Algorithm

From the series, you can see that any number in the series is a sum of the previous two numbers, i.e. $\text{number}_n = \text{num}_{n-1} + \text{number}_{n-2}$

Main Program

```
1. Get the number of Fibonacci numbers from the user.
2. If the number is <= 0, then
      a. Display cannot generate numbers
3.  else
      a. for I in 1 .. Number
            i. Call the recursive function Fibo
           ii. Display Value returned by Fibo
```

Fibo Function

```
1. If num < 2
      a. Return 1
2. Else
      a. Return (Fib(n-1)+ Fib(n-2));
```

Code Listing

```
GNAT 3.13p  (20000509) Copyright 1992-2000 Free Software Foundation, Inc.

Compiling: c:/docume~2/ joeb /desktop/16070/codeso~1/fibo_with_recursion.adb (source file time stamp:
2003-10-02 14:42:40)
```

```
---------------------------------------------------
-- Program to find N fibonacci numbers using
-- recursion.
-- Programmer : Joe B
-- Date Last Modified : October 01, 2003
---------------------------------------------------


with Ada.Text_Io;
with Ada.Integer_Text_Io;

procedure Fibo_With_Recursion is
   function Fibo (
         Num : Integer )
     return Integer is
   begin
      if (Num < 2) then
         return 1;
      else
         return(Fibo(Num-1) + Fibo(Num-2));
      end if;
   end Fibo;

   Number : Integer;

begin
   -- get the number from the user
   Ada.Text_Io.Put("Please Enter The Number Of Fibonacci Numbers : ");
   Ada.Integer_Text_Io.Get(Number);
   Ada.Text_Io.Skip_Line;

   if Number <=0 then
      Ada.Text_Io.Put("Cannot Generate Required Numbers");

   else
      for I in 1 .. Number loop
         Ada.Integer_Text_Io.Put(Fibo(I));
      end loop;
   end if;

end Fibo_With_Recursion;
```

```
41 lines: No errors
```

Note: While we are generating the numbers recursively, we are not displaying them as they are generated. Can you guess why?

The recursive function Fibo uses two recursive calls within the program. If you trace the program, you will see that the values are computed twice by the recursive function. If you try to display the values inside the recursive function, the values will be repeated.