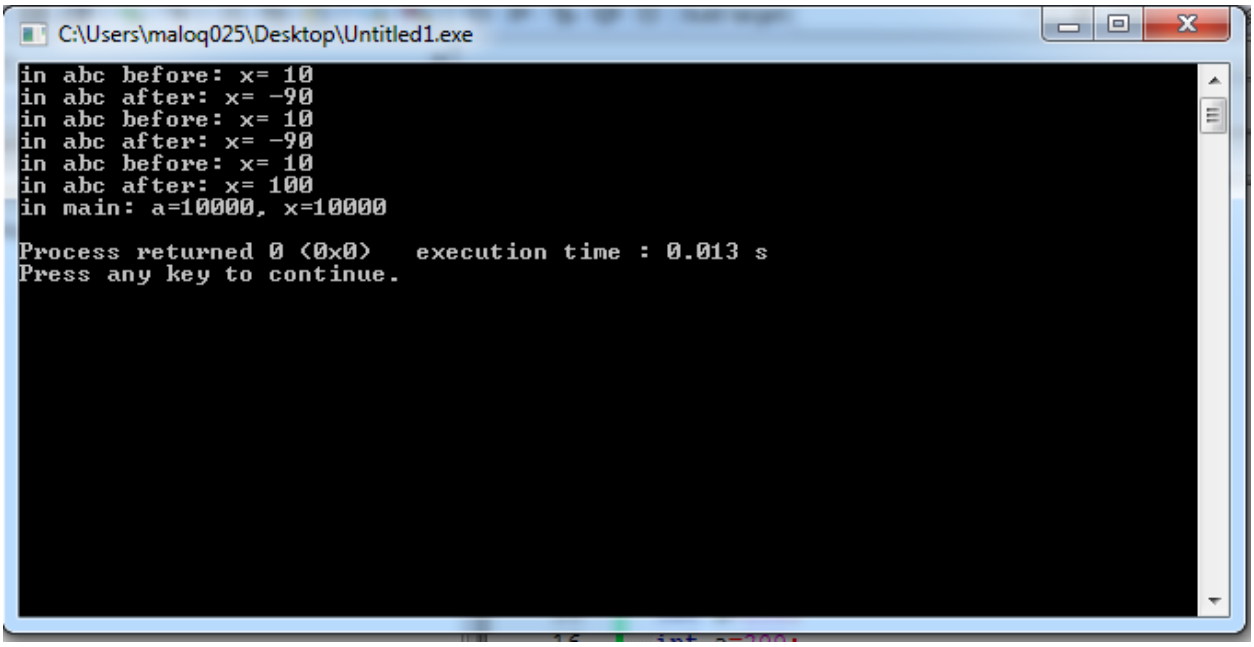
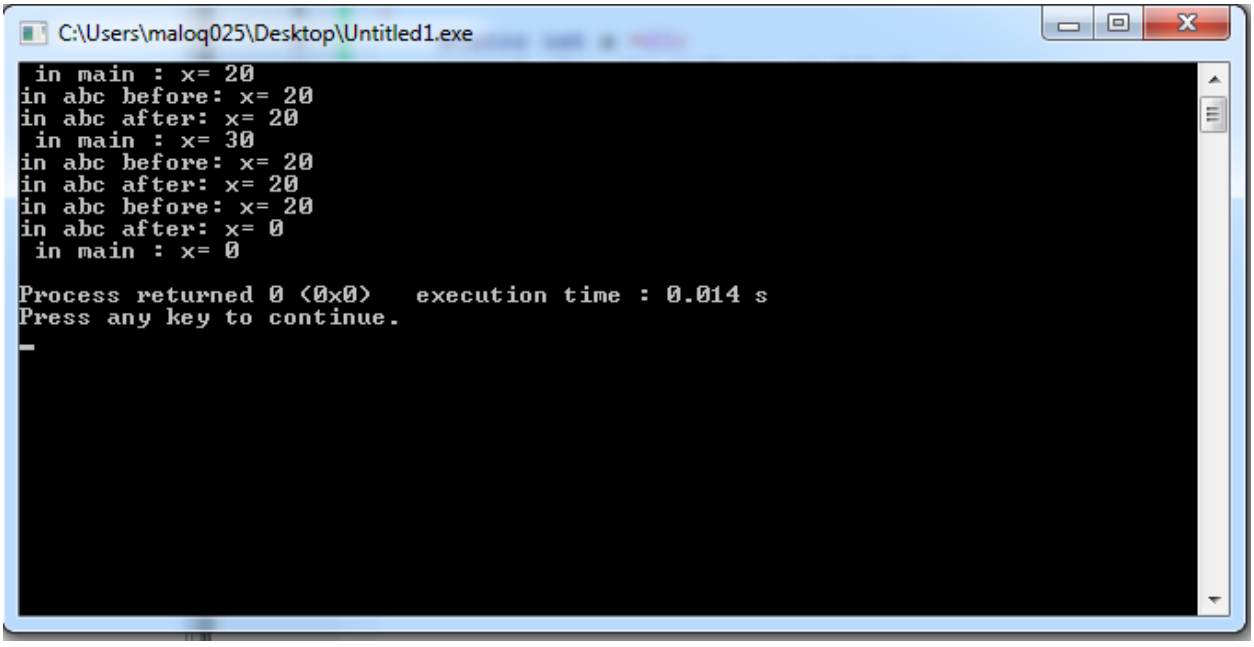
GNG1106 Homework Problem Set 3

Solutions

Question 1 (10 points): What will be the output of this program?



Question 2 (10 points): What will be the output of this program?



Question 3 (Programming) (40 points): The sequence 1; 1; 2; 3; 5; 8; 11; …. is called the Fibonnacci sequence and each number in the sequence is called a Fibonacci number. That is, the i^{th} Fibonacci number X_i is related to the $(i - 1)^{th}$ and $(i - 2)^{th}$ Fibonacci numbers X_{i-1} and X_{i-2} by $X_i = X_{i-1} + X_{i-2}$ and the first two Fibonacci numbers X_1 and X_2 are both 1. Write a function with the following prototype:

```
int getNextFibonacciNumber(void);
```

The function, when called for the i^{th} time, returns five terms starting from i^{th} to $(i^{th}+4)$ Fibonacci number.

Solution:

Steps from 1-6 should be included. Here is the programming code:

```
#include <stdio.h>

int  intial = 0, Last = 1, Next, i=0;

int Fibonacci_Seq()
{
      if ( i <= 1 )
         Next = 1;

      else
        { //Swap
         Next = intial + Last;
         intial = Last;
         Last = Next;
        }
}

void main()
{
    int integer;

    printf("Please enter your integer:\n");
    scanf("%d", &integer);
    printf("\nThe next five elements in the sequence are:\n");

    while (1)
    {
               if (i<integer)
            {
                Fibonacci_Seq();
                i++;
            }
```

```
        else if (i>=integer , i < (integer+5))
            {
                Fibonacci_Seq();
                printf("%d\n", Next);
                i++;
            }

}}
```

Question 4 (Programming) (40 points): Write a program that prompts the user to enter a positive integer N>=3 and prints the set of all binary ({0; 1}-valued and the matching Hex decimal values) strings. For example, if the user enters 3, the program prints

Binary	Hex
0 0 0	0
0 0 1	1
0 1 0	2
0 1 1	3
1 0 0	4
1 0 1	5
1 1 0	6
1 1 1	7

and if the user enters 4, the program prints

Binary	Hex
0 0 0 0	0
0 0 0 1	1
0 0 1 0	2
0 0 1 1	3
0 1 0 0	4
0 1 0 1	5
0 1 1 0	6
0 1 1 1	7
1 0 0 0	8
1 0 0 1	9
1 0 1 0	A
1 0 1 1	B
1 1 0 0	C
1 1 0 1	D
1 1 1 0	E
1 1 1 1	F

Solution:

Steps from 1-6 should be included. Here is the programming code:

```
#include<stdio.h>
#include<math.h>

int binary(int N, int i)
{
int K = pow(2,N);
 printf("%d",i/K);
if(N>=1)
 binary(N-1,i%K);
}

void main()
{
int i,N,limit;
int decimalNumber,remainder,quotient;
    int k=1,j,temp;
    char hexadecimalNumber[100];

 printf("Please etnter your decimal:\n");
 scanf("%d",&N);
 limit = pow(2,N);

 printf("Binary are:\t\tHex:\n");


for(i=0;i<limit;i++)
{
 quotient = i;
  k=1;

  printf("\n");
  binary(N-1,i);


 printf("\t\t\t");
    while(quotient!=0)
    {
         temp = quotient % 16;
      if( temp < 10)
           temp =temp + 48;
      else
         temp = temp + 55;
```

```
      hexadecimalNumber[k++]= temp;
      quotient = quotient / 16;
  }

if(i==0)
    printf("0");
    for(j = k -1 ;j> 0;j--)
    {

         printf("%c",hexadecimalNumber[j]);
    }

}
}
```