Name: Kunal Goyal
CS-800
Student Id: 10471054

Program to Calculate Nth Fibonacci number

This program will calculate Nth Fibonacci number. “N” should be read from the keyboard. The program will accept the input number then will check its Fibonacci value for particular index. for reference, please find below table.

$n =$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	...
$x_n =$	0	1	1	2	3	5	8	13	21	34	55	89	144	233	377	...

we can pass any number here and according to that it will show the result.

Fibonacci's sequence is characterized by the fact that every number after the first two is the sum of the two preceding ones. For Example:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 ,....so on

Program Execution Steps:

We are performing following steps to execute and build the logic in program.

We will Initialize the stack frame and after that we will pass the code for user input via keyboard. This input will be the index value for which we want to find Fibonacci number. This number would be passed to stack and then we will call the function which will calculate the nth Fibonacci number.

We will have a function called fibbNum, which will work in 2 parts. one is working for 0 and 1 and other part is part2 which will work for other inputs.

Program code screenshot:

EASy68K Editor/Assembler v5.16.01 - [untitled1.x68]

File Edit Project Options Window Help

```
*-----------------------------------------------------------------------------------------------------------
* Title      :   Calculate Nth Fibonacci number
* Written by :   Kunal Goyal
* Date       :   05/03/2021
* Description:   This program will calculate Nth Fibonacci number. "N" should be read from the keyboard.
*                The program will accept the input number then will check its Fibonacci value for particular index
*-----------------------------------------------------------------------------------------------------------
    ORG    $1000
Start:

LOOP    LEA     MESSAGE,A1
        move.b  #14,D0
        trap    #15

        * Stack frame initialization
        movea.l #1,a2

        * Read number from keyboard
        move.b #4,d0
        trap    #15

        * Move it to stack
        move.l d1,-(sp)

        * Call the function
        bsr    fibbNum

        * Pop the fib from the stack and store to fib
        adda.l #4,sp
        move.l d0,fib

       * Print the result
        move    #3,d0
        move.l fib,d1
        trap    #15

        SIMHALT

* Declare variables
```

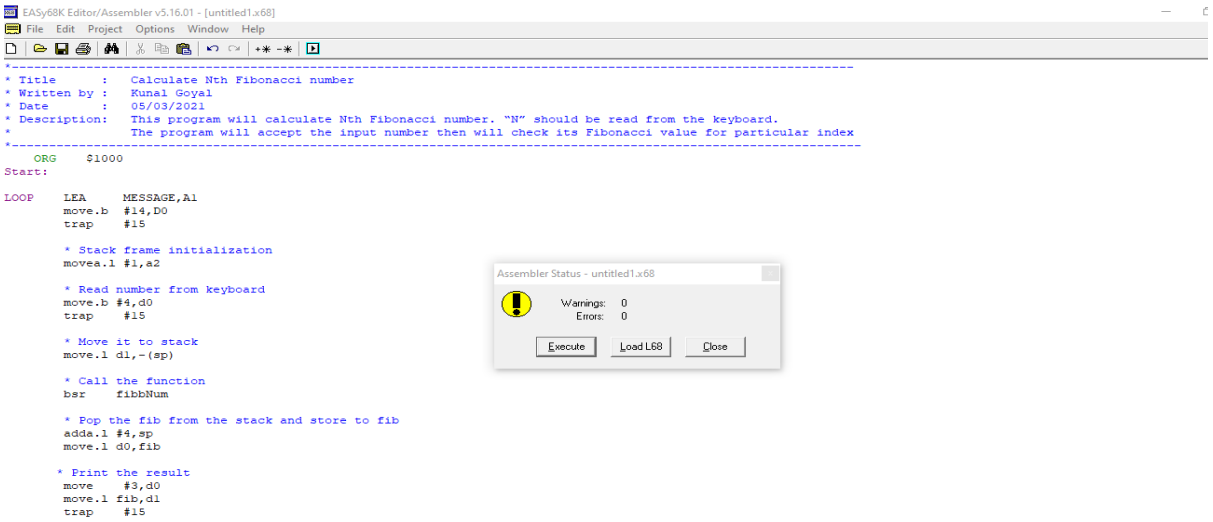
```
* Declare variables
fib: ds.l 1


fibbNum:

       * Allocating space and frame
        move.l a2,-(sp)
        move.l sp,a2
        suba.l #4,sp

       * Move number to d0
        move.l 8(a2),d0

       * Check number is 0
        cmp.l #0,d0

       * if not 0 branch to part1
        bne    part1

       * Return 0 if number is 0
        move.l #0,d0

       * Deallocation
        move.l a2,sp

       * Restore
        move.l (sp)+,a2

       * Return
        rts

part1:
       * Mone number to do
      move.l 8(a2),d0
```

```
        * Check number is 1
         cmp.l #1,d0

        * Branch to part 2 if not equal
         bne    part2

        * Return 1 if number is 1
         move.l #1,d0

        * Deallocate
         move.l a2,sp

        * Restore
         move.l (sp)+,a2

        * Return
         rts

part2:

        * Move number to d0
         move.l 8(a2),d0

        * Subtract one from the number
         sub.l #1,d0

        * Push it to the stack
         move.l d0,-(sp)

        * Compute fib(number-1)
         bsr    fibbNum

        * Pop number-1 from the stack
         adda.l #4,sp

        * Store the result of fib(number-1) to d0
         move.l d0,-4(a2)
```

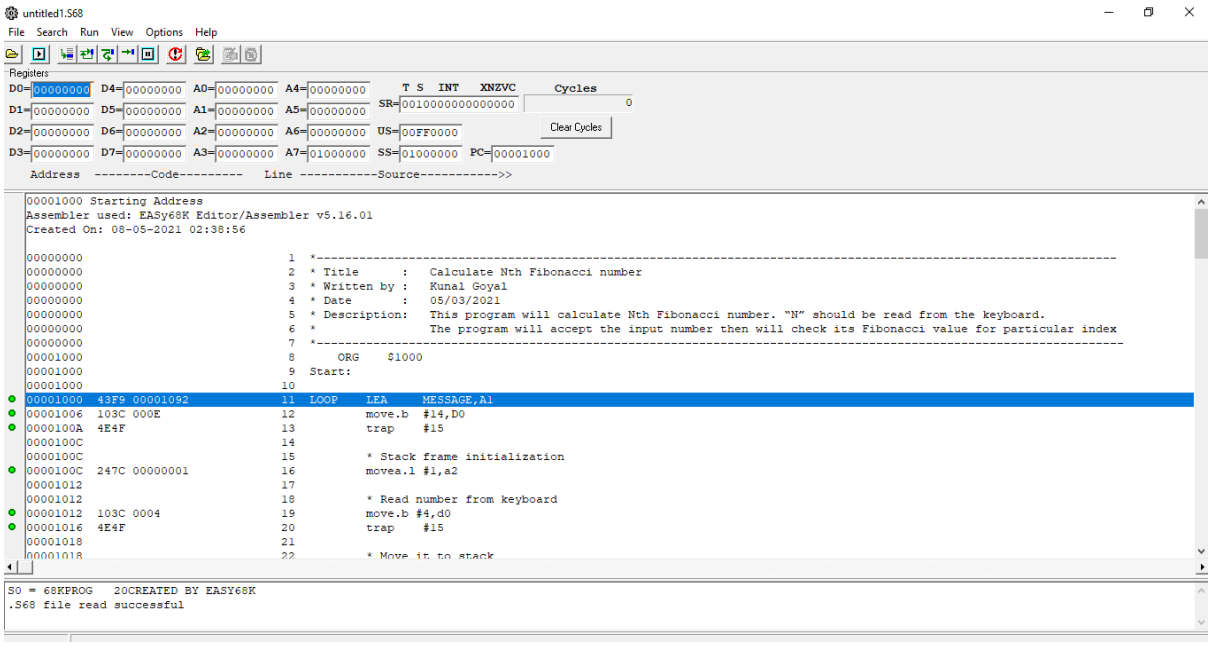
```
         bsr    fibbNum

        * Pop number-1 from the stack
         adda.l #4,sp

        * Store the result of fib(number-1) to d0
         move.l d0,-4(a2)

        * Move number to d0
         move.l 8(a2),d0

        * Subtract one from the number
         sub.l #2,d0

        * Push it to the stack
         move.l d0,-(sp)

        * Compute fib(number-2)
         bsr    fibbNum

        * Pop number-1 from the stack
         adda.l #4,sp

        * Compute fib(number-1)+fib(number-2)
         add.l -4(a2),d0

        * Deallocate
         move.l a2,sp

        * Restore
         move.l (sp)+,a2

         rts

CR      EQU     $0D
LF      EQU     $0A

MESSAGE DC.B    'Enter a index value between 0 and 20 to Get Fib Number: ',0
```

Error Count while debugging:

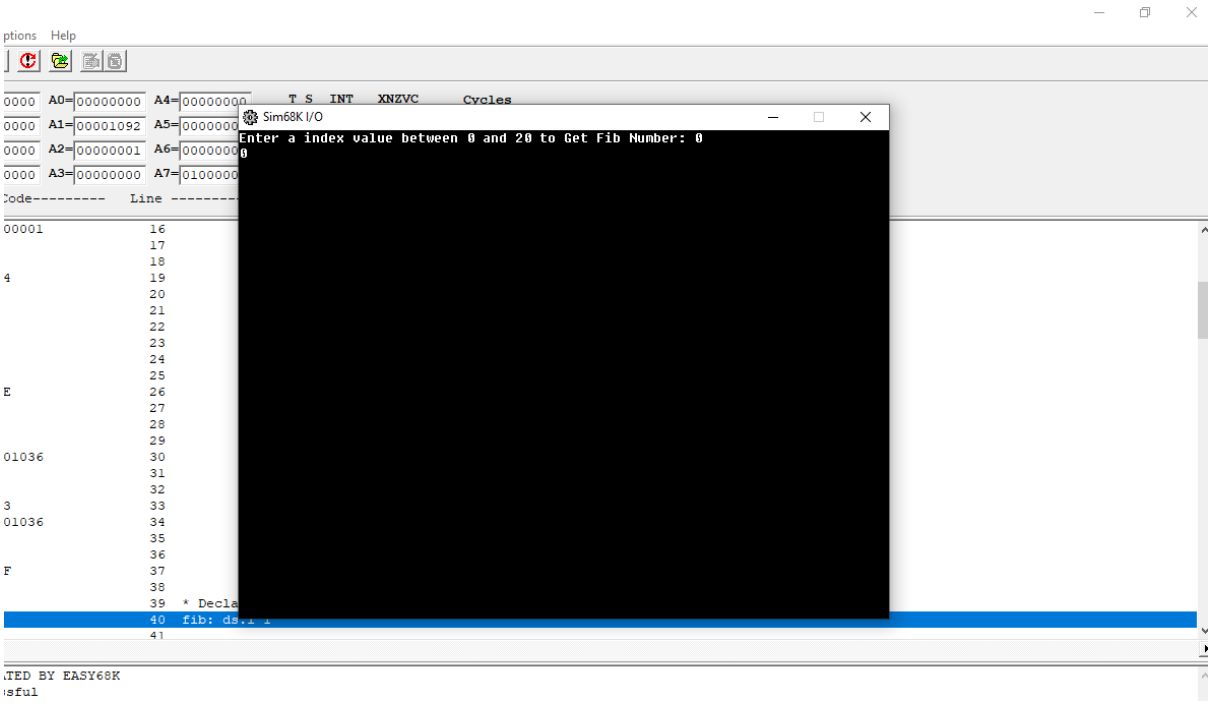
Total number or errors: 0
Total numbers of warning: 0



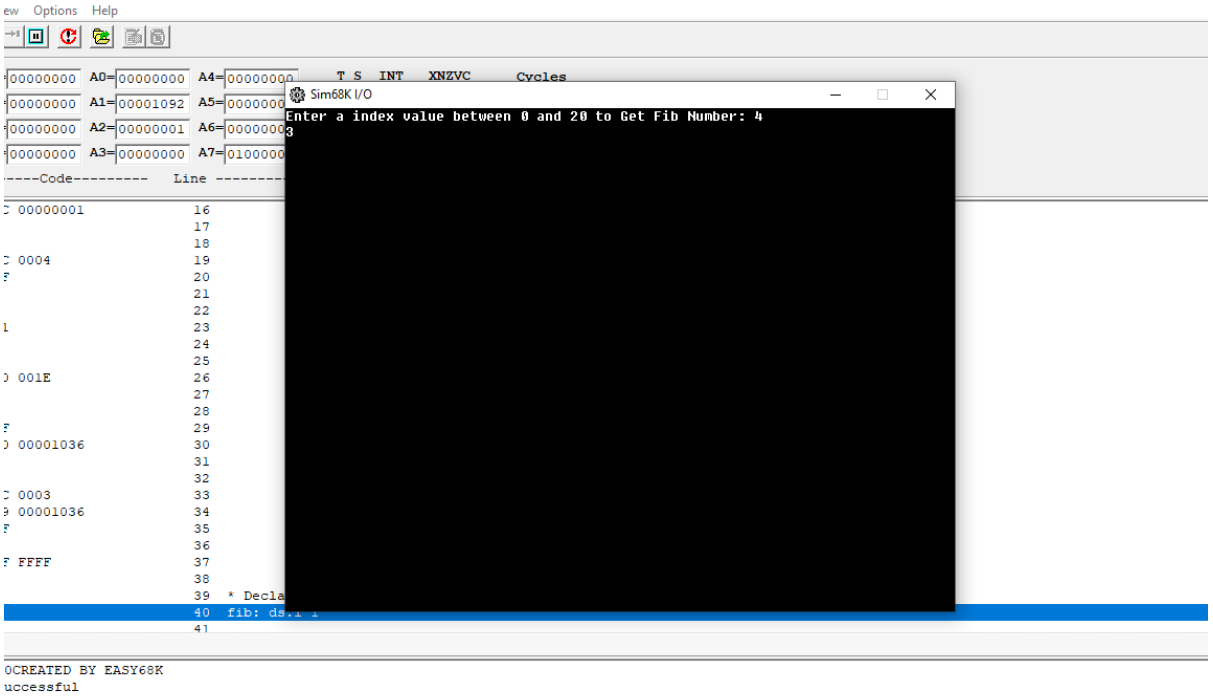
Execute Screen:



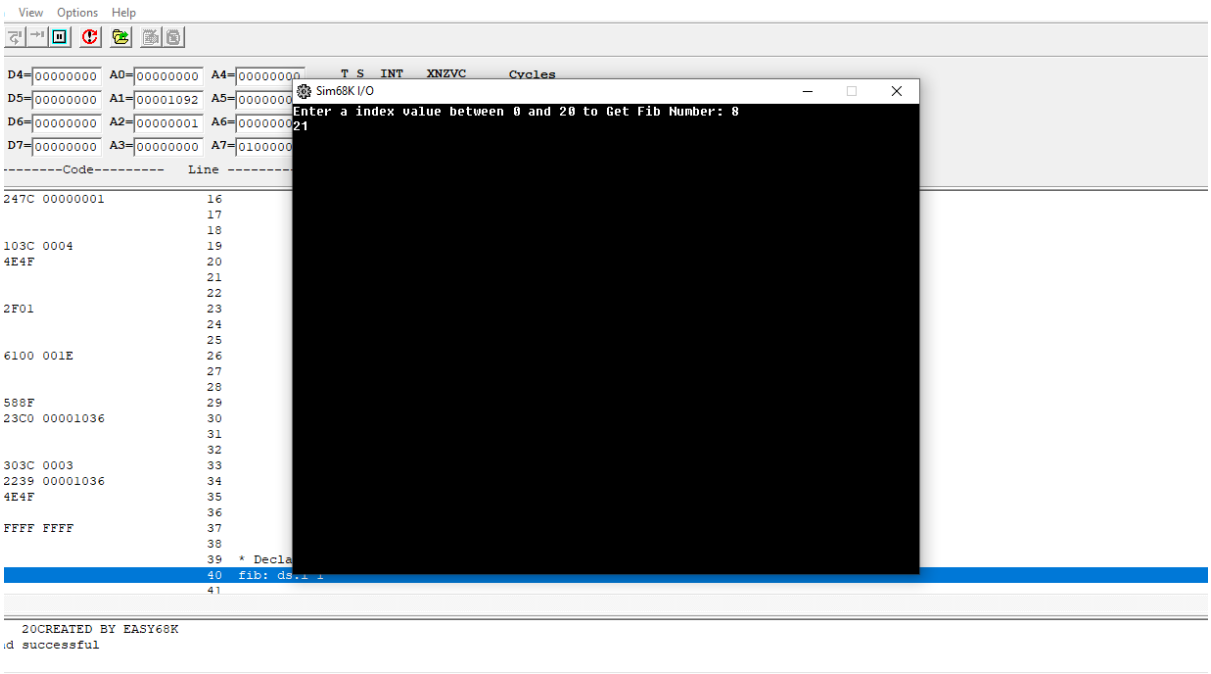
Output Screenshots:



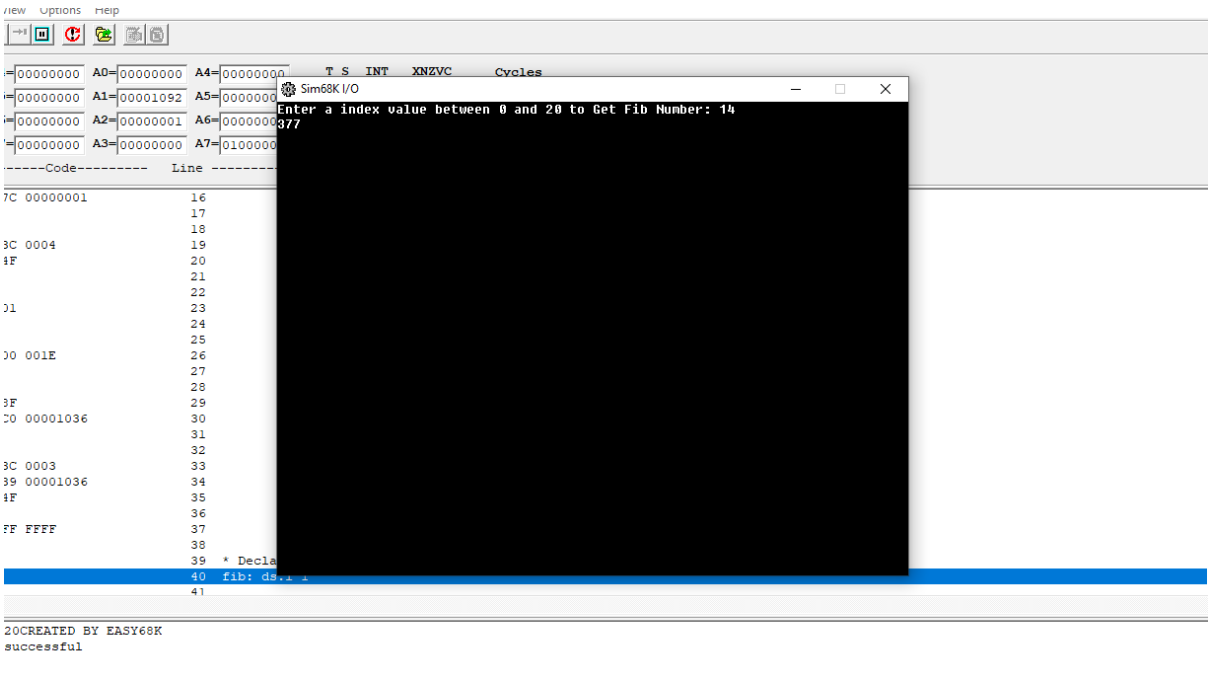
Output for index value 0



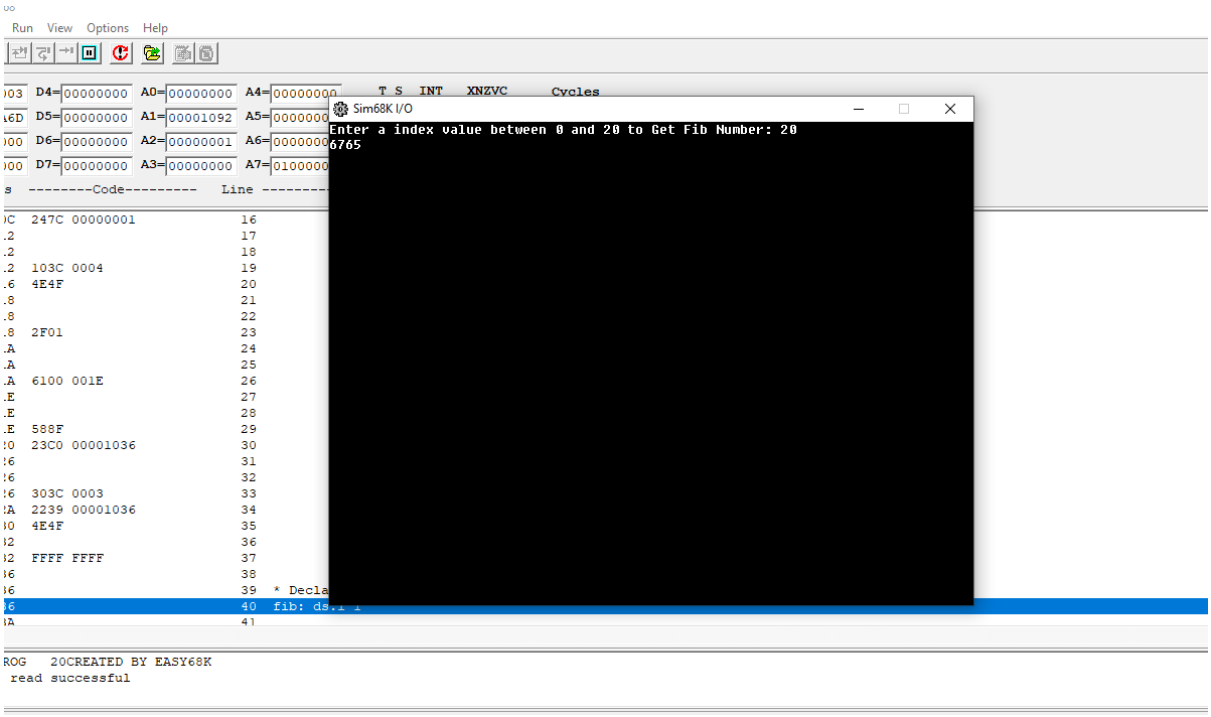
Output for Index Value 4



Output for Index value 8



Output for Index value 14



Output for index Value 20

Reference:

https://www.youtube.com/watch?v=RLXnBQtNaT0
https://www.youtube.com/watch?v=65_0jHDfpQ0
https://www.youtube.com/watch?v=-wHGbemFT4c
https://www.oreilly.com/library/view/microprocessor-theory-and/9780470380314/13_ch06.html
http://meseec.ce.rit.edu/eecc250-winter99/250-final-review.pdf
http://meseec.ce.rit.edu/eecc250-winter99/250-12-2-99.pdf