PRF 192

Workshop 4

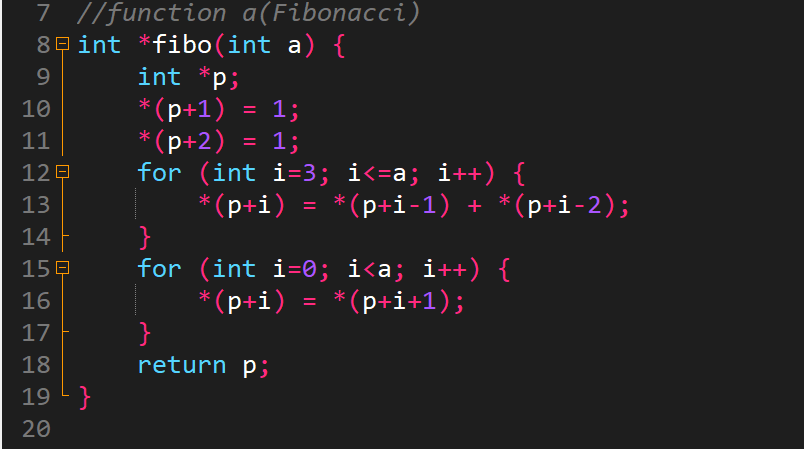
Name: Hoang Thuy Nguyen

ID: DE191056

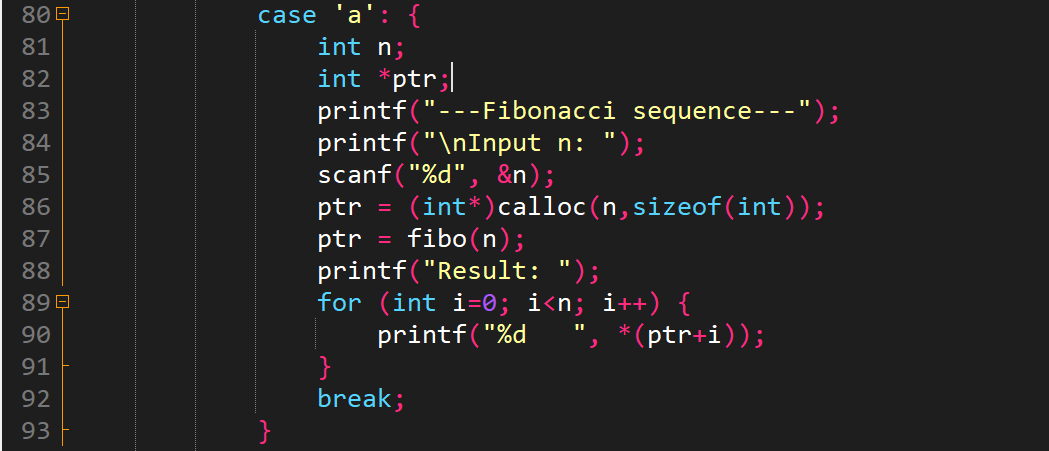
1. **Press “a” to generate an array containing n integers where each element is the sum of previous two integers.**

**Example: input 7, the output is 1 1 2 3 5 8 13**

1. Function:

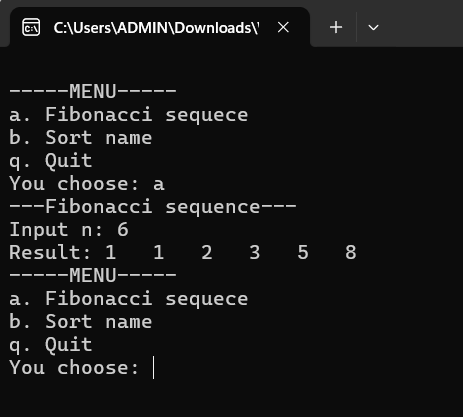


1. Call the function in main function:



1. Test:

* Case 1:



Walkthrough:

Line 85: enter 6→n=6

Line 87: pass n=6 into “fibo” function

Line 8: a = 6

Because after executing this program , we will receive an array of numbers with the principles of the problem math so I use pointer instead of array because I can use dynamic memory allocation for it(Line 86)

The principle of the problem math is that the next number is equal to the sum of the two previous numbers

Line 10: assign 1 to \*(p+1) ← the second element of array

Line 11: assign 1 to \*(p+2) ← the third element of array

Line 12: build for loop , let i run from 3 to a(a=6)

When i = 3 → \*(p+3) = \*(p+3-1) + \*(p+3-2) = 1 + 1 = 2

I = 4 → \*(p+4) = \*(p+4-1) + \*(p+4-2) = 1 + 2 = 3

…

Line 15: this loop will delete the first element, it mean \*(p) from array

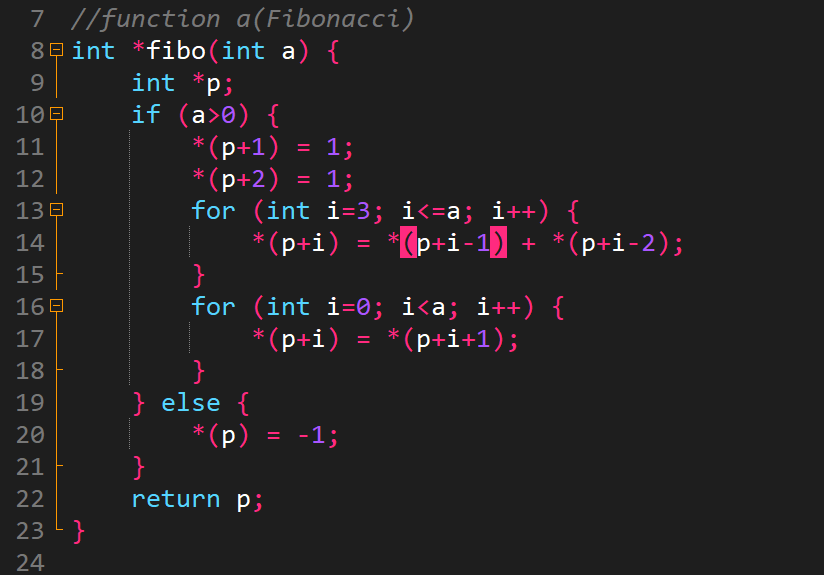
Line 18: function return pointer p

Line 87: assign value of pointer p of “fibo” function into pointer ptr of main function

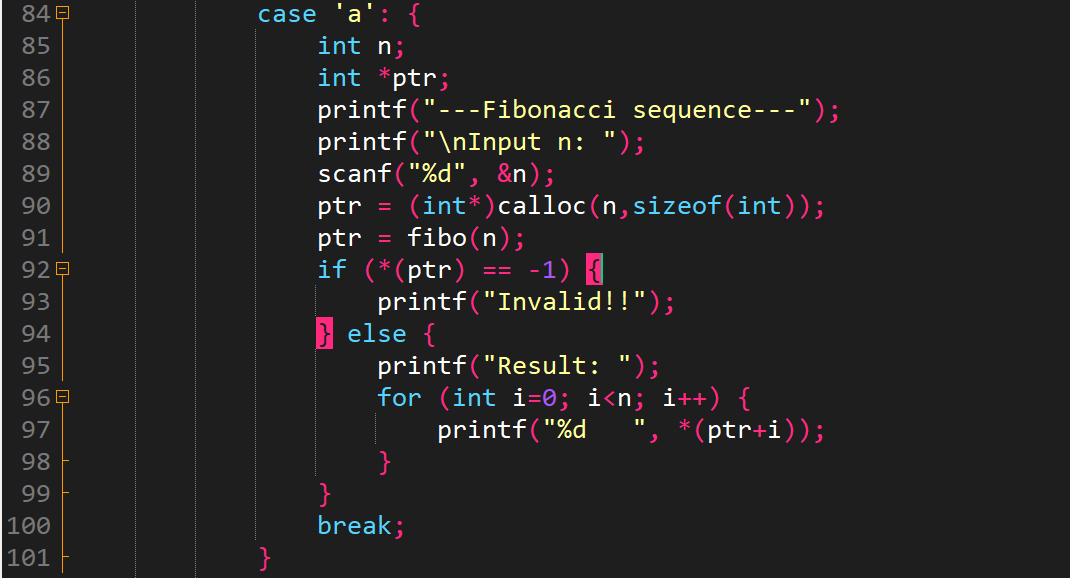
Line 89 → line 91: print the result on screen

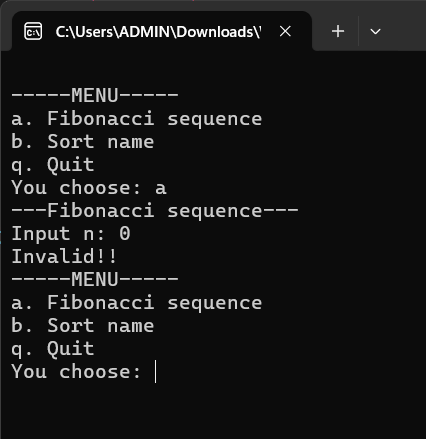
* Case 2:

The function:



Call function in the main:





Walkthrough:

Line 89: enter 0 → n=0

Line 90: dynamic memory allocation for pointer ptr

Line 91: pass n into fibo function

Line 8: n=0 → a=0

Because a=0 → Line 19: assign -1 to \*(p)

Line 22: the function return pointer p

Line 91: assign result of fibo function to pointer ptr of main

Because (p) = -1 →\*(ptr) = -1 →Line 93: print out on screen “Invalid!!”

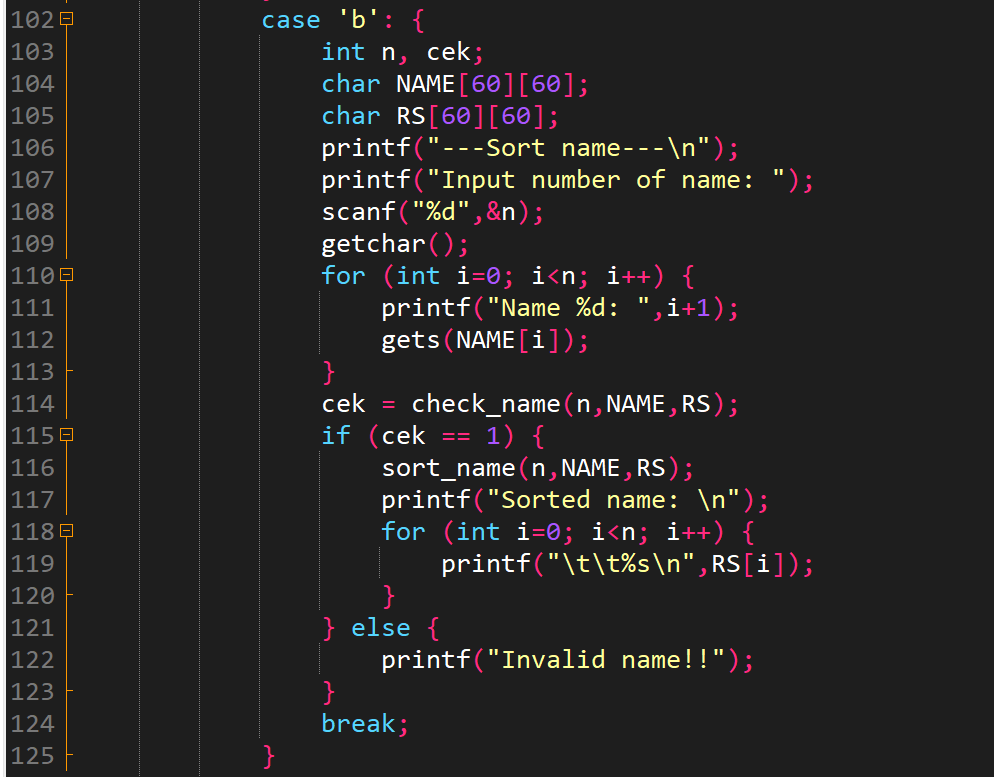
1. **Press “b” to sort a given array of names to increasing alphabet. Test your code in main function.**

**Example: nameArray = {“Nguyen A”, “Le C”, “Doan E”, “Tran B”, “Huynh D”}, then array after sorted ={“Nguyen A”, “Tran B”, “Le C”, “Huynh D”, “Doan E”}**

1. Function:

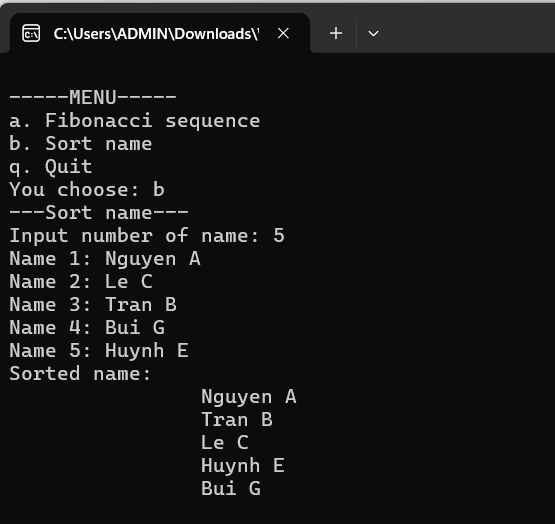


1. Call the function in main function:



1. Test:

* Case 1:



Walkthrough:

Line 108: enter 5 → n=5

Line 109: Getchar function to consume “enter” character

At here, to store a name (including first name, middle name and last name) we use 2D array, so at line 104 and line 105 we declare NAME and RS with 2D array

With NAME array, we use to enter name from user

With RS array, is used to store sorted names

Line 110: for loop is used to input string name

Line 116: n = 5 , NAME and RS 2D array to sort\_name function

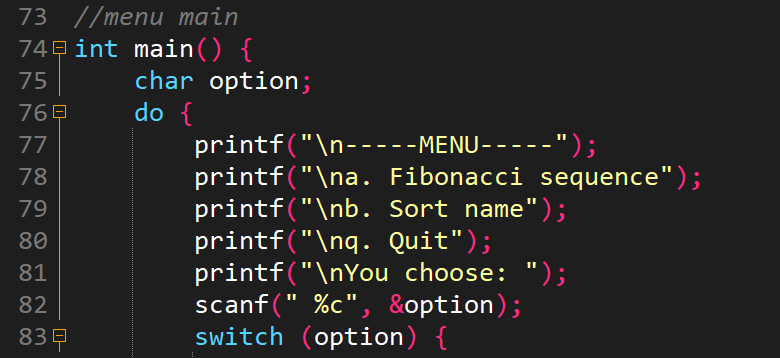
Line 42: a=5, name 2D array store NAME array, temp\_sen 2D array store RS array

Line 47 → line 55: use j and i loop to store the first letter of each name (only name. Example: Nguyen Lan Anh → store “A”, Phan Thiet → store “T”). To store the first letters, I used pointer “temp” (with role is like normal array). Simultaneously, store entire name(including last, middle, first name) into “temp\_sen”, use strcpy to perform it

Line 57→line 71: continue to use to i and j loop to sort the first letters in pointer “temp” in ascending order, and at that time swap name elements in “temp\_sen”.

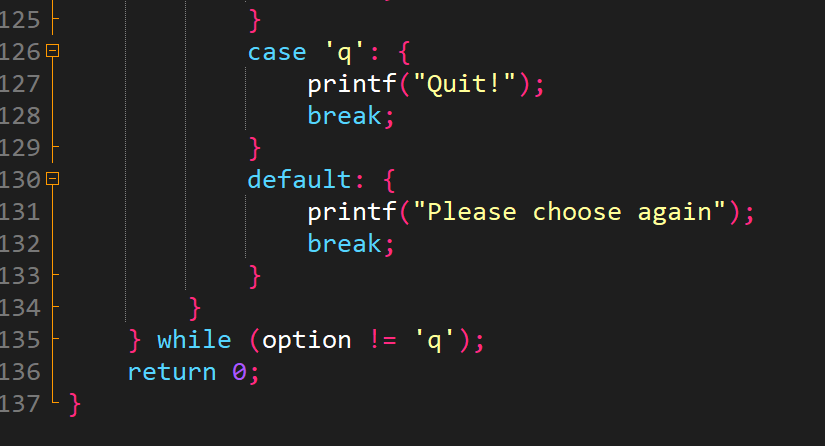
Line 116: sorted name in “temp\_sen” array will be stored in RS array

Line 118 → line 119: print sorted names on screen

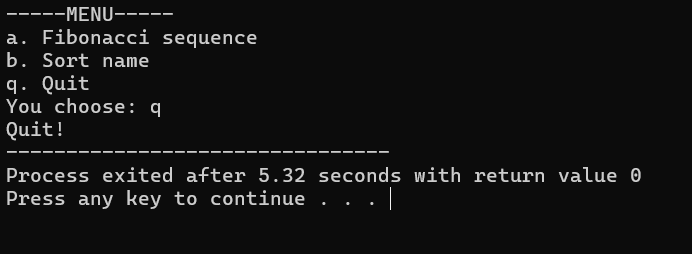
1. 

Line 83: enter any character to choose program which you want to computer execute

If you want to exit program you must enter ‘q’ 🡪exit. Unless the program will loop



Case you want to exit:



Case you enter wrong character:

