C++ Programming

(335)

REGIONAL – 2016

Production Portion:

Program 1: Fibonacci / Perfect Numbers ______________ (300 points)

TOTAL POINTS ______________ ***(300 points)***

Judge/Graders: Please double check and verify all scores and answer keys!

NOTES TO CONTESTANT:

You will have 90 minutes to complete your work.

Your name and/or school name should *not* appear on any work you submit for grading.

Copy your entire solution/project to the flash drive provided. You must submit your entire solution/project so that the graders may open your project to review the source code. You must ensure that the files required to run your program are present and will execute on the flash drive provided. Note that the flash drive letter may *not* be the same when the program is graded as it was when you created the program. It is recommended that you use relative paths rather than absolute paths to ensure that the program will run regardless of the flash drive letter. The graders will **not** compile or alter your source code to correct for this. Submissions that do *not* contain source code will **not be graded**.

Development Standards:

- Standard name prefixes must be utilized for variables.
- All subroutines, functions, and methods must be documented with comments explaining the purpose of the method, the input parameters (if any), and the output (if any).

Your application will be graded on the following criteria:

Solution and Project

The project is present on the flash drive ____ 10 points
The project is named according to the naming conventions ____ 10 points

Program Execution

Code copied to USB drive and the program runs from USB ____ 20 points

If the program does *not* execute, then the remaining items in this section receive a score of zero.

The program correctly displays the menu options ____ 20 points
The program correctly displays the first 20 Fibonacci numbers ____ 40 points
The program correctly displays all Perfect numbers under 1000 ____ 40 points
The program clears the screen before the menu is re-displayed ____ 10 points

Source Code Review

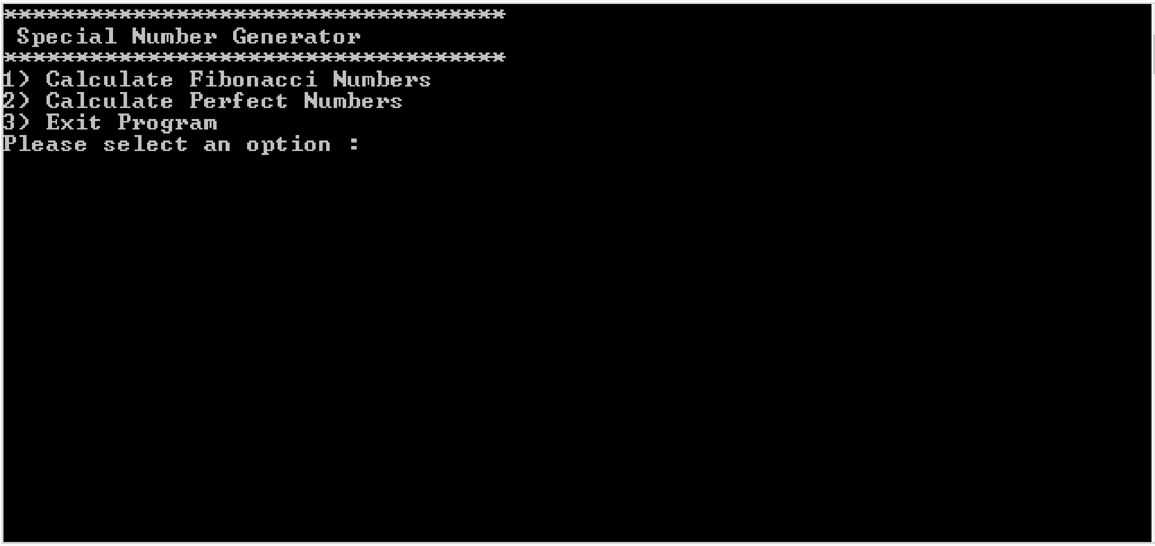
Code is commented at the top, for each function, and as needed ____ 15 points
Code uses reasonable and consistent variable naming conventions ____ 15 points
A method called, “generateFibonacciNumbers” is implemented ____ 50 points
A method called, “generatePerfectNumbers” is implemented ____ 50 points
A menu is programmatically created ____ 10 points
Incorrect menu option choice is properly handled ____ 10 points

Total Points: ______/300 points

Generating Fibonacci Numbers and Perfect Numbers

Write a program that will display a menu with options to calculate Fibonacci Numbers and Perfect Numbers. The menu should provide three options (See Figure 1):

- Calculate First 20 Fibonacci Numbers
- Calculate All Perfect Numbers Less Than 1000
- Exit the Program



```
***********************************
 Special Number Generator
***********************************
1) Calculate Fibonacci Numbers
2) Calculate Perfect Numbers
3) Exit Program
Please select an option :
```

Figure 1

<u>Definition of Fibonacci Numbers</u>

The Fibonacci Sequence is the series of numbers:

1, 1, 2, 3, 5, 8, 13, 21, 34, …

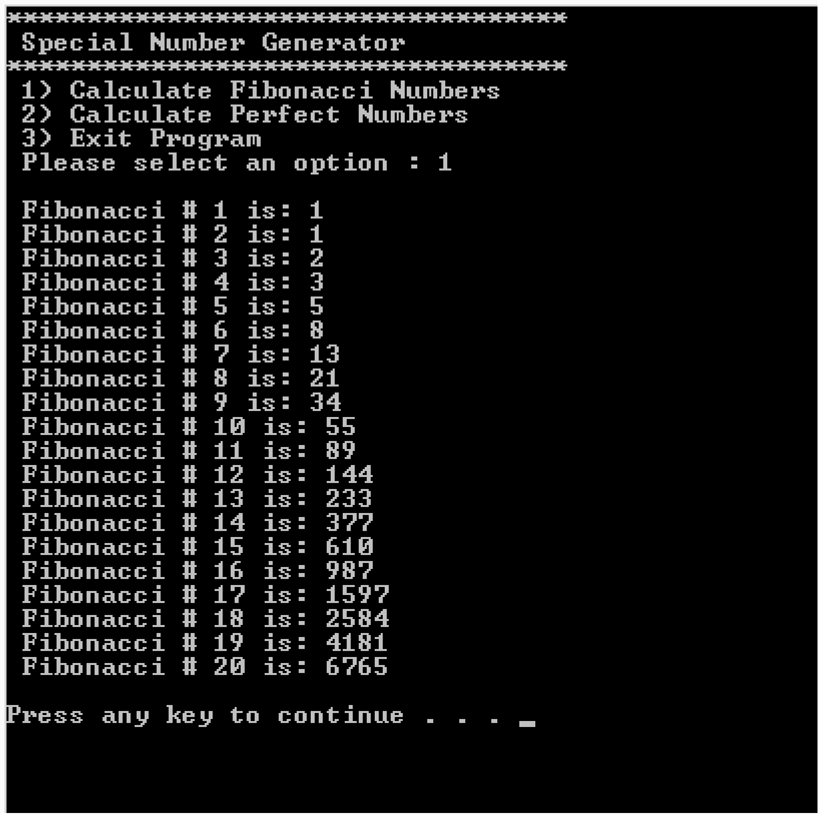
The next number is found by adding up the two numbers before it.

<u>Definition of a Perfect Number</u>

A perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself. For example: 6 is a perfect number, because the divisors of 6 are 1, 2, and 3. The sum of 1 + 2 + 3 equals 6.

Requirements:

1. You must create a C++ console application named CPP_335_ContestantNumber, where ContestantNumber is your BPA assigned contestant number (including dashes). For example, CPP_335_01_2345_6789.
2. Your name and contestant number must appear as a comment at the top of the main source code file.
3. Create a function called, "generateFibonacciNumbers" that performs the calculations and generates the output of the first 20 Fibonacci Numbers.
    a. The function output should be formatted like Figure 2.
4. Create a function called, "generatePerfectNumbers" that performs the calculations and generates the output of the all perfect numbers less than 1000.
    a. The function output should be formatted like Figure 3.
5. The program should run until the user enters the exit number of 999.
6. The program should display a message if the user enters an invalid menu choice (see Figure 4).
7. Clear screen each time the menu is redisplayed.

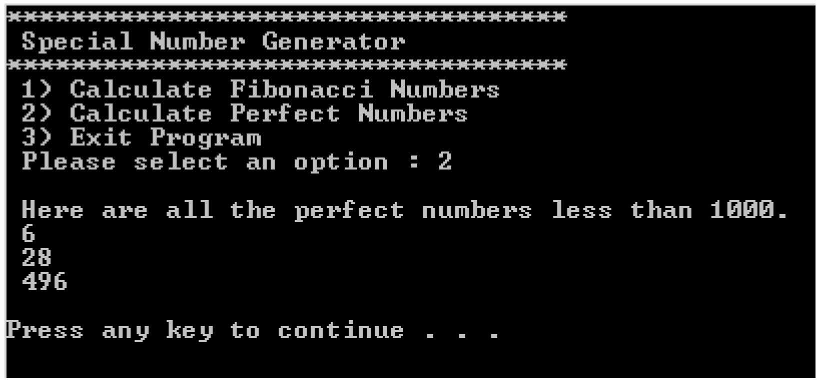


```
***********************************
 Special Number Generator
***********************************
 1) Calculate Fibonacci Numbers
 2) Calculate Perfect Numbers
 3) Exit Program
 Please select an option : 1

 Fibonacci # 1 is: 1
 Fibonacci # 2 is: 1
 Fibonacci # 3 is: 2
 Fibonacci # 4 is: 3
 Fibonacci # 5 is: 5
 Fibonacci # 6 is: 8
 Fibonacci # 7 is: 13
 Fibonacci # 8 is: 21
 Fibonacci # 9 is: 34
 Fibonacci # 10 is: 55
 Fibonacci # 11 is: 89
 Fibonacci # 12 is: 144
 Fibonacci # 13 is: 233
 Fibonacci # 14 is: 377
 Fibonacci # 15 is: 610
 Fibonacci # 16 is: 987
 Fibonacci # 17 is: 1597
 Fibonacci # 18 is: 2584
 Fibonacci # 19 is: 4181
 Fibonacci # 20 is: 6765

Press any key to continue . . . _
```

Figure 2

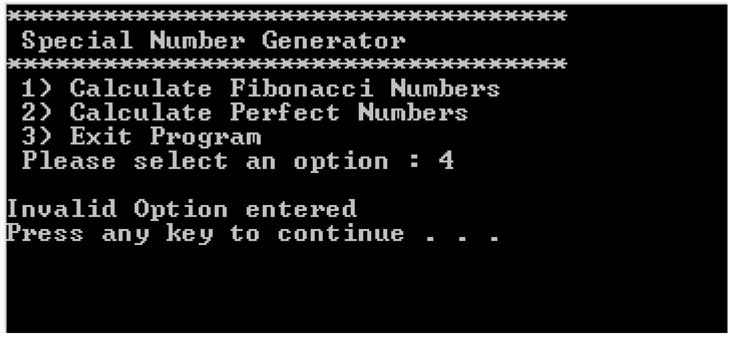


```
***********************************
 Special Number Generator
***********************************
 1) Calculate Fibonacci Numbers
 2) Calculate Perfect Numbers
 3) Exit Program
 Please select an option : 2

 Here are all the perfect numbers less than 1000.
 6
 28
 496

Press any key to continue . . .
```

Figure 3



```
***********************************
 Special Number Generator
***********************************
 1) Calculate Fibonacci Numbers
 2) Calculate Perfect Numbers
 3) Exit Program
 Please select an option : 4

Invalid Option entered
Press any key to continue . . .
```

Figure 4

NOTE TO GRADER:
This is just a sample source code. Contestant code may differ.
****
MAIN
****

```
// Regional2015FibonacciPerfectNumbers.cpp
//This program displays a menu to generate Fibonacci numbers and perfect numbers.
#include <iostream>
using namespace std;

void generateFibonacciNumbers();
void generatePerfectNumbers();

int main()
{
       int option; // user's entered option will be saved in this variable
       do //do-while loop starts here.that display menu again and again until user select to
exit program
       {
              //Displaying Options for the menu
              cout << "***********************************\n";
              cout << " Special Number Generator\n";
              cout << "***********************************\n";
              cout << " 1) Calculate Fibonacci Numbers " << endl;
              cout << " 2) Calculate Perfect Numbers " << endl;
              cout << " 3) Exit Program " << endl;

              //Prompting user to enter an option according to menu
              cout << " Please select an option : ";
              cin >> option;  // taking option value as input and saving in variable
"option"

              if(option == 1) // Checking if user selected option 1
              {
                     //Call Fibonacci Method
                     generateFibonacciNumbers();
              }
              else if(option == 2) // Checking if user selected option 2
              {
                     //Call Perfect Number Method
                     generatePerfectNumbers();
              }
              else if(option == 3) // Checking if user selected option 4
              {
                     cout << "Terminating Program" << endl;
              }
              else //if user has entered invalid choice (other than 1,2,3 or 4)
              {
                     //Displaying error message
                     cout << "Invalid Option entered" << endl;
              }
       }
       while(option != 3);  //condition of do-while loop

       return 0;
}
```

```
void generateFibonacciNumbers()
{
       cout << "\n";
       //Calculate and output the first 20 Fibonacci numbers.  (Assume it starts with 1).
       int max, current, first = 1, second = 1;
       double next;
       max = 20;

       for ( current = 0 ; current < max ; current++ )
       {
              if ( current <= 1 )
                     next = 1;
              else
              {
                     next = first + second;
                     first = second;
                     second = next;
              }
              cout << " Fibonacci # " << current + 1 << " is: " << next << endl;
       }
       cout << "\n";
       system("pause");
       system("cls");

}

void generatePerfectNumbers()
{
       int i = 1, sum = 0;

       cout << "\n";
       cout << " Here are all the perfect numbers less than 1000.\n";

       //Output all perfect numbers less than 1000.
       for (int num = 1; num < 1000; num++)
       {
              while(i < num)
              {
                     //Check for factors of current number, if factor is found, add it to
the sum
                     if(num % i == 0)
                     {
                            sum = sum + i;
                     }

                     i++;
              }

              if(sum == num)
              {
                     cout << " " << num << endl;
              }

              i = 1; //reset i
              sum = 0; //reset sum to check next number
```

```
        }
        cout << "\n";
        system("pause");
}
```