


National University of Computer and Emerging Sciences, Lahore Campus

	Course Name:	Programming Fundamentals	Course Code:	CS 118
	Program:	BS(CS,SE,DS)	Semester:	Fall 2020
	Duration:	90 Minutes	Total Points:	40
	Paper Date:	26/11/2020	Page(s):	7
	Exam Type:	Midterm-II	Section:	ALL

Registration No. _____

Instructions:

Attempt all questions

You might use extra sheets for working but please try to write the final answer in the space provided for it

Problem No 1:

[2 + 4 + 2 + 4 Points]

What gets printed when each of the following code segmented is executed

Program Segment	Output
<pre>for(int num = 1; num<= 5; num++) cout<<num<<" "<<num*(num+1)/2<<endl;</pre>	<pre>1 1 2 3 3 6 4 10 5 15</pre>
<pre>int main() { int P[20]; P[0] = 0; P[1] = 0; for(int i = 2; i < 20; i++) P[i] = i; for(int i = 2; i< 5; i++){ for(int j = 2*i; (P[i] != 0) && (j< 20); j+=i) P[j] = 0; } for(int i = 0; i< 20; i++){ if(P[i] != 0) cout<< P[i]<<endl; } return 0; }</pre>	<pre>2 3 5 7 11 13 17 19</pre>

<pre> int num = 10, i = 0; while(i < num){ if(i%2 == 0) cout<<i/2<<"-"; else cout<<num-i/2-1<<"-"; i++; } </pre>	<div>0-9-1-8-2-7-3-6-4-5-</div>
<pre> #include <iostream> using namespace std; int waitWhat(int& a, int& b) { int c = 0; a = c++; b = (a++) + c; return 2 * a + b; } int hellNo(int& a, int b, int& c) { c *= 2; int i = 0; for (i = 1; i <= c; i++){ a = a + c; c = c - 1; } b = c + a; return i; } int main() { int a = 1, b = 2, c, d; c = waitWhat(a, b); cout << "A: " << a << endl; cout << "B: " << b << endl; cout << "C: " << c << endl << endl; d = hellNo(a, c, b); cout << "A: " << a << endl; cout << "B: " << b << endl; cout << "C: " << c << endl; cout << "D: " << d << endl; return 0; } </pre>	<div> A: 1 B: 1 C: 3 A: 3 B: 1 C: 3 D: 2 </div>

Problem No 2:**[12 Points]**

Finding the position of largest and smallest values within a sequence a_0, a_1, \dots, a_{N-1} of N numbers is a useful operation.

Write a program that take as input a value N followed by reading N numbers and print the **first occurrence of the smallest value** and **last occurrence of the largest value** in the sequence.

NOTE: You are **not allowed to use arrays** for solving this problem. **Furthermore**, you must **input the values only once**

Following are some sample input along with the corresponding sample outputs

SAMPLE INPUT	SAMPLE OUTPUT
10 1 2 3 4 5 6 7 8 9 10	Smallest Value is at Position 0 Largest Value is at Position 9
6 12 3 1 3 12 1	Smallest Value is at Position 2 Largest Value is at Position 4
5 2 2 2 2 2	Smallest Value is at Position 0 Largest Value is at Position 4

WRITE THE PROGRAM ON THE NEXT PAGE

BLANK PAGE FOR WRITING ANSWER

```
int main() {
    int N = 0;
    int value = 0;

    cout << "Enter Number N: ";
    cin >> N;

    if (N > 0) {
        int maxValue = 0, maxPosition = 0;
        int minValue = 0, minPosition = 0;

        cout << "Enter N values:" << endl;
        cin >> value;
        maxValue = value;
        minValue = value;

        for (int i=1; i<N; i++) {
            cin >> value;

            if (value < minValue) {
                minValue = value;
                minPosition = i;
            }

            if (value >= maxValue) {
                maxValue = value;
                maxPosition = i;
            }
        }
        cout << "Smallest Value is at Position " << minPosition << endl;
        cout << "Largest Value is at Position " << maxPosition << endl;
    }
    else
        cout << "Error: N is Smaller than or Equal to 0" << endl;
    return 0;
}
```

Problem No 3:**[12 + 4 Points]**

A **positive integer** is said to be a **SQUARE FREE** if it is not divisible by any perfect square other than 1. Following are examples of some square free numbers 1, 2, 3, 5, 6, 7, 10, 11, 13, 14, 15, 17... whereas the numbers 4, 8, 9, 12... are not square free

- Write a function that will display all square free numbers that are less than N where N is a parameter of the function. The function **must also return** the **count of** these **square free** numbers to the calling function.
- Also write the **main** function that uses the function in **part a)** to display the square free numbers that are less than a certain value M specified by the user. The **main** function must also display the **count of square-free numbers that are less than N** and the **count of non-square-free numbers that are less than N**.

Following are some sample Inputs and the corresponding output that must be produced by your program

SAMPLE INPUT	SAMPLE OUTPUT
10	1, 2, 3, 5, 6, 7 Square Free Count = 6 Non-Square Free Count= 3
6	1, 2, 3, 5 Square Free Count = 4 Non-Square Free Count= 1
4	1, 2, 3 Square Free Count = 3 Non-Square Free Count= 0

NOTE: YOU MIGHT WRITE ADITIONAL FUNCTION(s) IF NEEDED

BLANK PAGE FOR WRITING ANSWER

//Solution without Helper Function

```
int printSquareFreeNumbers(int N) {
    int sqFreeCount = 0 ;
    bool flag = true;
    for (int i = 1; i < N; i++) {
        flag = true;
        for (int j = 2; j <= i; j++) {
            if ((i % (j * j)) == 0)
                flag = false;
        }
        if (flag)
        {
            cout << i << ",";
            sqFreeCount++;
        }
    }
    cout << endl;
    return sqFreeCount;
}

int main() {
    int M=0;
    cout << "Enter the Number M: ";
    cin >> M;
    if (M > 1) {
        int sqFree = printSquareFreeNumbers(M);
        cout << "Square Free Count = " << sqFree << endl;
        cout << "Non-Square Free Count= " << M - sqFree - 1 << endl;
    }
    else
        cout << "Error: M is Smaller than or Equal to 1" << endl;
    return 0;
}
```

//Solution with Helper Function

```
bool isSquareFree(int i) {
    bool flag = true;
    for (int j = 2; j <= i; j++) {
        if ((i % (j * j)) == 0)
            flag = false;
    }
    return flag;
}

int printSquareFreeNumbers(int N) {
    int sqFreeCount = 0;
    for (int i = 1; i < N; i++) {
        if (isSquareFree(i))
        {
            cout << i << ",";
            sqFreeCount++;
        }
    }
    cout << endl;
    return sqFreeCount;
}
```

```
int main() {  
    int M=0;  
    cout << "Enter the Number M: ";  
    cin >> M;  
    if (M > 1) {  
        int sqFree = printSquareFreeNumbers(M);  
        cout << "Square Free Count = " << sqFree << endl;  
        cout << "Non-Square Free Count= " << M - sqFree - 1 << endl;  
    }  
    else  
        cout << "Error: M is Smaller than or Equal to 1" << endl;  
    return 0;  
}
```