National University of Computer and Emerging Sciences, Lahore Campus

Course:	PF Lab	Course Code:	CL-118
Program:	BS (Computer Science)	Semester:	Fall 2019
Duration:	150 Minutes	Total Marks:	60(30+30)
Paper Date:	27 Nov 2019	Weight	40%
Section:	C, D, E, F, K, L	Page(s):	2
Exam:	Final Term	Reg. No	

Instruction/Notes:

1. Understanding the question paper is also part of the exam, so do not ask any clarification.
2. No USB's, PHONES and INTERNET are allowed.
3. Talking/Discussion is not allowed. It is your responsibility to protect your code and save it from being copied. If you don't protect it all matching codes are considered copy/cheating cases.
4. Create a folder as you roll number. e.g **l191234** and put both source (.cpp files) inside folder and submit on following submission path: **\\cactus\Xeon\Fall 2019\Shakeel Zafar\PF Final Exam # 1\Section X**. Where X is your section.

Question # 1: [Find a Pattern/Substring]

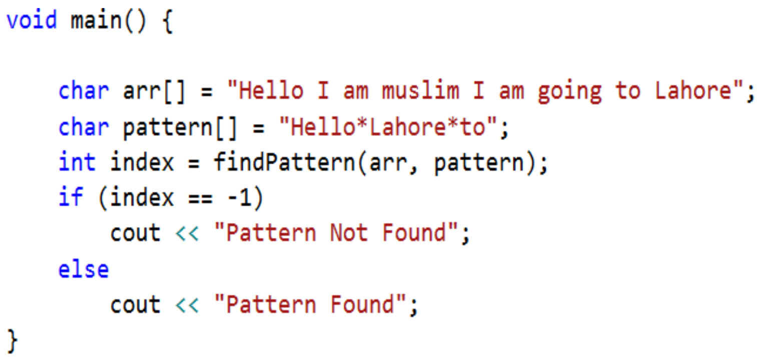
You need to write a c++ program which have a character array and a pattern to be searched.
You need to implement these functions in order to meet the given sample output.

```
int findPattern (char str [], char pattern []);
```

This function receives the array str and finds the pattern in str. This function makes a new character array to store a word/substring from pattern (assume a word in pattern is separated by *). When a word/substring is formed, it calls the function findSubString with the array (str), substring and an index value, from where the search should begin.

```
int findSubString (char arr [], char sub [], int index);
```

This function receives an array of characters and a substring to be searched and an index value, from where the search will start. This function returns the index where the substring is found else it returns -1.



```
void main() {

    char arr[] = "Hello I am muslim I am going to Lahore";
    char pattern[] = "Hello*Lahore*to";
    int index = findPattern(arr, pattern);
    if (index == -1)
        cout << "Pattern Not Found";
    else
        cout << "Pattern Found";
}
```

Note:

Star **(*) in pattern** denotes that there may exist zero or more characters between the words separated by (*), **in str**. Pattern may contain more than one stars (*). Each word is separated by single space **in str**.

Sample I/O:

Str:	Hello I am Muslim I am going to Lahore
Pattern:	Hello*Lahore*to

Output: Pattern Not Found.

Str:	Hello I am Muslim I am going to Lahore
Pattern:	Hello*am*to

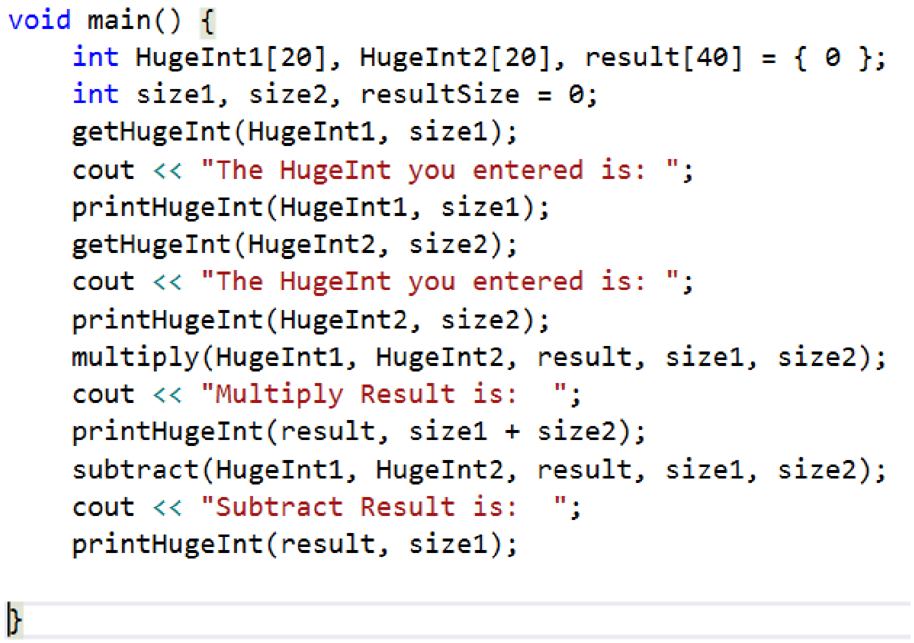
Output: Pattern Found.

Question # 2: [HugeInt Arithmetic Operations]

In this question, you have to implement some operations for **HugeInt**. Which can handle integers up to (maximum) 20 decimal digits. **HugeInt is an array of integer digits.** On a single index there will be only one digit. You need to implement these functions in order to meet the given sample output.

```
void getHugeInt (int HugeInt [], int &size);
//asks the user for size and takes input in HugeInt digit by digit.

void printHugeInt (int HugeInt [], int size);
//skipping the leading zeros, it prints the HugeInt.

void multiply (int HugeInt1[], int HugeInt2[], int result [], int s1, int s2);
//This function receives two HugeInt arrays and multiply them and store the result in result array.
//The size you need to consume for result array is (s1 + s2).


void subtract (int HugeInt1[], int HugeInt2[], int result [], int s1, int s2);
This function receives two HugeInt arrays and subtract HugeInt2 from HugeInt1 and store the result in
result array. The size you need to consume in result array is (s1). Note. Always the s1 will be
greater than s2.

You are given a main function and a sample input/output.
```

```
Sample I/O:
Enter the size of HugeInt: 3
        Enter digit # 1 of Number:  1
        Enter digit # 2 of Number:  0
        Enter digit # 3 of Number:  1
The HugeInt you entered is: 101

Enter the size of HugeInt: 1
        Enter digit # 1 of Number:  2
The HugeInt you entered is: 2

Multiply Result is:  202
Subtract Result is:  99
```



```
void main() {
    int HugeInt1[20], HugeInt2[20], result[40] = { 0 };
    int size1, size2, resultSize = 0;
    getHugeInt(HugeInt1, size1);
    cout << "The HugeInt you entered is: ";
    printHugeInt(HugeInt1, size1);
    getHugeInt(HugeInt2, size2);
    cout << "The HugeInt you entered is: ";
    printHugeInt(HugeInt2, size2);
    multiply(HugeInt1, HugeInt2, result, size1, size2);
    cout << "Multiply Result is:  ";
    printHugeInt(result, size1 + size2);
    subtract(HugeInt1, HugeInt2, result, size1, size2);
    cout << "Subtract Result is:  ";
    printHugeInt(result, size1);

}
```

-------------------------------- **BEST OF LUCK ☺** -------------------------------------