# Lab # 1: Review of C++: Pointers

**Subject:** **Data Structures and Algorithms**

**Course:** **BESE15 B**

**Date:** **30th September,10**

---

**EXERCISES:**

**Pointers 1**: A function that compares 2 strings and returns **true** if they are the same string, **false** if they are not.

```
#include <iostream>
#include <conio>
int compStr(char*,char*);
void main (void)
{
char str1[30]  ;
char str2[30]   ;
cout<<"Enetr first string:  ";
cin.getline(str1,'$');
cout<<"Enetr second string:  ";
cin.getline(str2,'$');
//char str1[]="pakistan1";
//char str2[]="pakistan2";
if(compStr(str1, str2)==1)
   cout<<"True";
else
   cout<<"False";
   getche();
}
int compStr(char *str1, char *str2) {
char *str1Ptr = str1;
char *str2Ptr = str2;
   int i = 0;
     while (*str1Ptr) {
               if (*str1Ptr++ == *str2Ptr++)
         i=1;
       else{ i=0;  break; }
       if(*str1Ptr=='\0' && *str2Ptr=='\0')
       i=1;
       else
       i=0;    }
               return i;
 }
```

**Pointers 2:** using structures and pointers.

```
#include<iostream>
#include<conio>
int main()
{
  struct candybar
  {
    char bar_name[15];
    float weight;
    short calories;
  };
  candybar snacks[3] = {{"Mocha Munch", 2.3, 350},
                        {"Cocoa Crush", 4.3, 920},
                        {"Tobler", 3.7, 512}};
  candybar *p = snacks;
  for (; p < snacks+3; p++)
    cout << "Bar: " << p->bar_name
    << ", Weight: " << p->weight
    << ", Calories: " << p->calories << endl << endl;
    getche();
  return 0;
}
```

**TASK:**

1. Write a function that combines two character arrays and stores the result in a third array using pointers.

2. Write a program that takes input from a user and stores it in a dynamically created array through pointers and prints it in the correct and reverse order.

3. Write a program that has a function for integer addition and one for integer subtraction. Create a third function 'operation' which calls these two functions using pointers-to-functions. Call 'operation' from main.
    a. Addition- input:int x, int y; output: int result
    b. Subtraction- input:int x, int y; output: int result
    c. Operation- input:int x, int y, pointer to function; output: int result

    *int* operation (*int* x, *int* y, *int* (*functocall)(*int*,*int*))
    z = (*functocall)(x,y);