

## Lab#9:

### Pointers

Total marks: 20 (10 each)

**Submission deadline:** 3<sup>rd</sup> May, 2020, Sunday till 11:59pm

**Submission guidelines:**

1. Put all your .cpp and relevant files in a folder. Zip that folder and name it with your roll number.
2. Email the zipped folder at [natalia@pucit.edu.pk](mailto:natalia@pucit.edu.pk). You must put the **subject** of your email as: **“Lab#9”**
3. Students who are facing issues with their laptop or compiler can write the code on paper and send the screenshots, following the above guideline. Since your issue will be genuine therefore relaxation will be given accordingly. Don't worry!

### Task 1: Pointers, pointer to pointer, and addresses

Consider the following code.

```
#include <iostream>
using namespace std;
int main()
{
    int var1 = 10, var2 = 5;
    int* p1, * q1, * r1;
    int** g;
    int var3;
    var3 = ++var2;

    p1 = &var1;
    q1 = &var2;
    r1 = &var3;

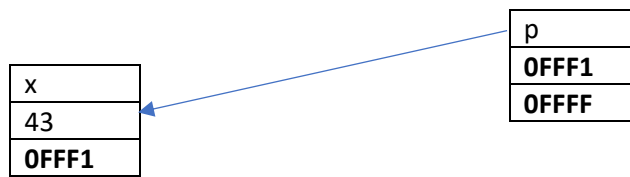
    *g = r1;

    *p1 = var3++;
    *q1 = ++var1;

    //add appropriate lines of code to print the things mentioned in the table below
    return 0;
}
```

Add the appropriate lines of code in the program given above. Print the following things and fill in the table. You need to **submit** following things:

1. Complete code .cpp file (used in printing addresses/values on the screen)
2. Following table filled with appropriate value (share its picture or write in MS Word as per your convenience).
3. Memory map diagram (made on paper or on MS Word). Mention the addresses and value you get after running complete code on your system. Draw it just as the map is drawn in our lecture slides. E.g. see the sample below in which a pointer p is pointing towards x.

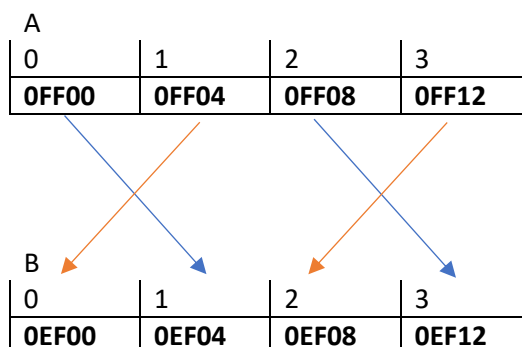


**For 2 and 3, the addresses will depend on your system's memory allocation therefore addresses will vary from one machine to another. Any cheating attempt will be identified quickly.**

Address of variable var1:	
Address of variable var2:	
Address of variable var3:	
Address of pointer variable p1:	
Address of pointer variable q1:	
Address of pointer variable r1:	
Value at location pointed by variable p1:	
Value at location pointed by variable q1:	
Value at location pointed by variable r1:	
Value at location pointed by variable g:	
Value at location pointed by the pointer that is pointed by g (i.e. contents of location indirectly pointed by g):	
Address of location pointed by the pointer that is pointed by g (i.e. contents of location indirectly pointed by g):	
Address of pointer variable g	

## Task 2: Array of pointers

In main(), write a logic that declares two array of pointers with sizes as shown below. Print the address of location of cells of both arrays. Pls note that contents of this array would not be integer values, instead there will be addresses. Your task is to create a following link between them. You can create a link / point locations to each other by using pointers concept. That is, A[0] is pointing towards B[1] and so on. At the end, display the contents of both the arrays.



## Optional (Just for practice, not graded)

### Task 3: Pointers and strings

---

Write a function ReverseOrder that declares a string and take an input in it. It then reverses its order and displays it. Use only pointer notation (you are not allowed to use array subscript [] notation in the function. You can use more than one pointers and strings in this program if you need any.

Void ReverseOrder();

**Example:**

i) Input: This is reverse order program

Output: program order reverse is This

ii) Input: This is exercise eight

Output: eight exercise is This