Submit one zipped folder named as your rollNumber that contains the required .cpp files and images of paper work.

Pointers

In C++, pointers are variables that store the memory addresses of other variables. Here is how we can declare pointers. Here, we have declared a pointer pointVar of the int type.

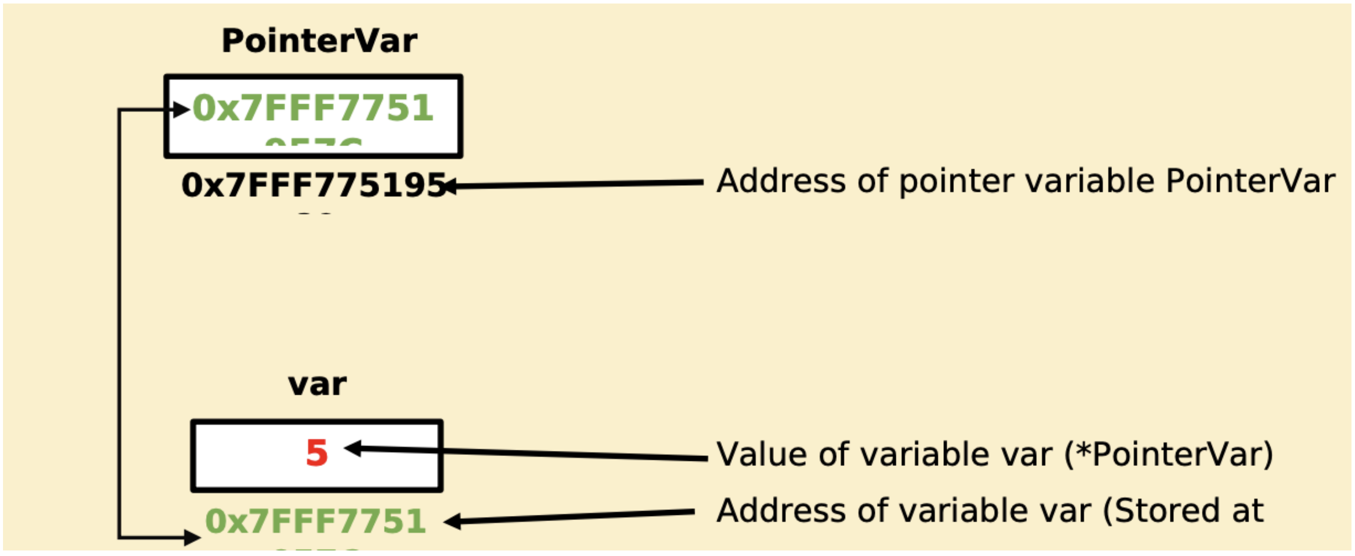
```
int *pointVar;
```

Assigning Addresses to Pointers

```
int *pointVar, var;
var = 5;
// assign address of var to pointVar pointer
pointVar = &var;
```

Here, 5 is assigned to the variable var. And, the address of var is assigned to the pointVar pointer with the code pointVar = &var.

Graphical Representation of Pointers


PointerVar
0x7FFF7751
0x7FFF775195
Address of pointer variable PointerVar
var
5
Value of variable var (*PointerVar)
0x7FFF7751
Address of variable var (Stored at

Get the Value from the Address Using Pointers

To get the value pointed by a pointer, we use the * operator.

```
int *pointVar, var;
var = 5;

// assign address of var to pointVar
pointVar = &var;

// access value pointed by pointVar
cout << *pointVar << endl;           // Output: 5
```

In the above code, the address of var is assigned to pointVar. We have used the *pointVar to get the value stored in that address. When * is used with pointers, it's called the dereference operator. It operates on a pointer and gives the value pointed by the address stored in the pointer. That is, *pointVar = var.

Changing Value Pointed by Pointer

If pointVar points to the address of var, we can change the value of var by using *pointVar. For example,

```
int var = 5;
int* pointVar;

// assign address of var
pointVar = &var;

// change value at address pointVar
*pointVar = 1;

cout << var << endl;   // Output: 1
```

Here, pointVar and &var have the same address, the value of var will also be changed when *pointVar is changed.

C++ Pointers and Arrays

In C++, Pointers are variables that hold addresses of other variables. Not only can a pointer store the address of a single variable, it can also store the address of cells of an array. Consider this example:

```
int *ptr;
int arr[5];

 // store the address of the first element of arr in ptr
ptr = arr;
```

Here, ptr is a pointer variable while arr is an int array. The code ptr = arr; stores the address of the first element of the array in variable ptr. Notice that we have used arr instead of &arr[0]. This is because both are the same.
The addresses for the rest of the array elements are given by &arr[1], &arr[2], &arr[3], and &arr[4].

Point to Every Array Elements

```
// ptr + 1 is equivalent to &arr[1]
// ptr + 2 is equivalent to &arr[2]
// ptr + 3 is equivalent to &arr[3]
// ptr + 4 is equivalent to &arr[4]
```

Similarly, we can access the elements using the single pointer.

```
// use dereference operator
    *ptr == arr[0];
// *(ptr + 1) is equivalent to arr[1];
// *(ptr + 2) is equivalent to arr[2];
// *(ptr + 3) is equivalent to arr[3];
// *(ptr + 4) is equivalent to arr[4];
```

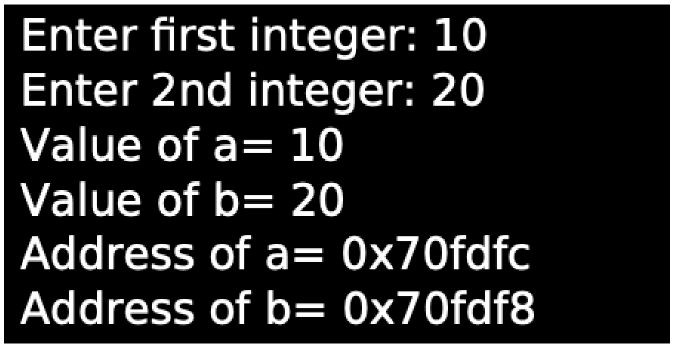
Suppose if we have initialized ptr = &arr[2]; then

```
// ptr - 2 is equivalent to &arr[0];
// ptr - 1 is equivalent to &arr[1];
// ptr + 1 is equivalent to &arr[3];
// ptr + 2 is equivalent to &arr[4];
```

Task 1

Write a program that asks the user to enter integers as inputs to be stored in the variables a and b respectively. Also create two integer pointers named ptrA and ptrB. Assign the addresses of a and b to ptrA and ptrB respectively. Display the values and addresses of a and b using ptrA and ptrB.

Sample Output (addresses will be different in your case)



```
Enter first integer: 10
Enter 2nd integer: 20
Value of a= 10
Value of b= 20
Address of a= 0x70fdfc
Address of b= 0x70fdf8
```

Task 2

& can be used as reference operator and address of operator. Write the output of the following code and check your output by retyping the following code on the compiler. Assume that address of a = 0x16b1fb068

```
cout << endl << "Using & as refernce operator" << endl;
a = 10;
int& ref = a;  // ref is a reference to a
ref = 20;      // Modifying ref also modifies a
cout << "Value of a: " << a << endl;

cout << endl << "Using & as address of operator" << endl;
a = 42;
cout << "Address of a: " << &a << endl;
cout << "Address of ref: " << &ref << endl;

cout << endl << "Assigning value of b" << endl;
ref = b;
cout << "Address of ref: " << &ref << endl;
cout << "value of ref: " << ref << endl;
cout << "value of a: " << a << endl;
```

Task 3

Write the output of the following code and check your output by retyping the following code on the compiler. Assume that address of a = 0x16b1fb068 and ptr = 0x16b1d3060

```
int a = 10;
int *ptr = &a;
cout << *ptr << endl;
cout << &a << endl;
cout << ptr << endl;
cout << *ptr << endl;

*ptr = 7;

cout << a << endl;
cout << &ptr << endl;
cout << *&a << endl;

a = 2;

cout << *&*&a << endl;
cout << &*ptr << endl;
cout << *&*ptr << endl;
```

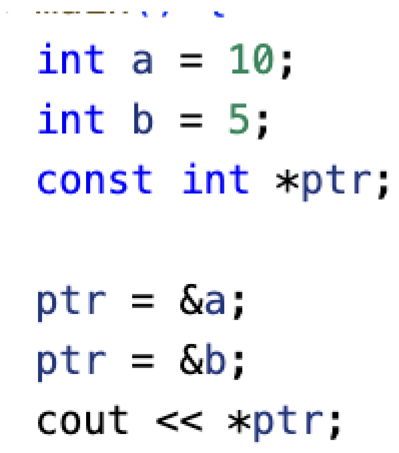
Task 4

Write the output of the following code and check your output by retyping the following code on the compiler. Assume that starting address of arr = 0x16d117050

```
int arr[5] = {1,21,3,4,5};
cout << arr << endl;
cout << *arr << endl;
cout << (arr+2) << endl;
cout << *(arr+2) << endl;
cout << *arr-4 << endl;
int *ptr = arr;
cout << *++ptr << endl;
cout << ++*ptr ;
```

Task 5

Write the output of the following code and check your output by retyping the following code on the compiler. In case of error, mention the line number

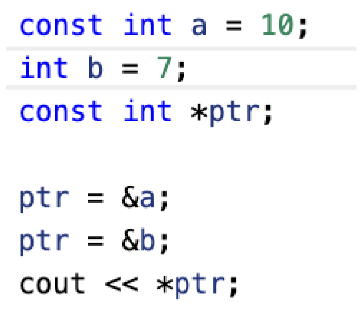


```
int a = 10;
int b = 5;
const int *ptr;

ptr = &a;
ptr = &b;
cout << *ptr;
```

Task 6

Write the output of the following code and check your output by retyping the following code on the compiler. In case of error, mention the line number

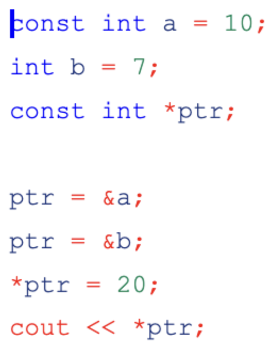


```
const int a = 10;
int b = 7;
const int *ptr;

ptr = &a;
ptr = &b;
cout << *ptr;
```

Task 7

Write the output of the following code and check your output by retyping the following code on the compiler. In case of error, mention the line number

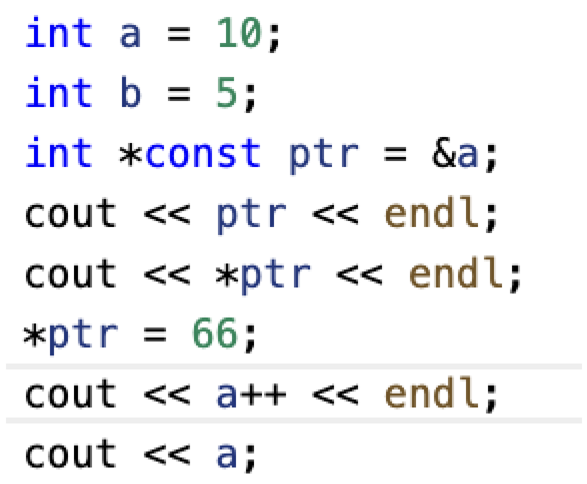


```
const int a = 10;
int b = 7;
const int *ptr;

ptr = &a;
ptr = &b;
*ptr = 20;
cout << *ptr;
```

Task 8

Write the output of the following code and check your output by retyping the following code on the compiler. In case of error, mention the line number



```
int a = 10;
int b = 5;
int *const ptr = &a;
cout << ptr << endl;
cout << *ptr << endl;
*ptr = 66;
cout << a++ << endl;
cout << a;
```

Task 9

Create a function that takes five pointers as arguments to find the sum of 5 numbers using pointers.

Task 10

Create a C++ program to swap the values of two variables. Arguments must be passed by reference.

Task 11

Create a C++ program to find the largest number from the array of 7 numbers using pointer. Subscript notation not allowed.