**Part 1**

#include<iostream>

using namespace std;

int\* GetArrayFromUser(int& size)

{

cout<<"\n\n\nI am inside GetArrayFromUser\n";

cout<<"Enter size of Array:\t";

cin>>size;

int\* tempArray = 0;

if(size > 0)

{

tempArray = new int[size];

for(int i = 0; i< size ; i++)

tempArray[i] = i+1;

}

cout<<"tempArray is pointing to location:\t"<<tempArray<<endl;

cout<<"Address of tempArray:\t"<<&tempArray<<endl;

return tempArray;

}

void UpdateArray(int\* intArray, const int& size)

{

cout<<"\n\n\nI am inside UpdateArray\n";

cout<<"intArray is pointing to location:\t"<<intArray<<endl;

cout<<"Address of intArray:\t"<<&intArray<<endl;

for(int i = 0; i< size ; i++)

intArray[i] = intArray[i] \* 2;

}

void DisplayArray(int\* tempArray, const int& size)

{

cout<<"\n\n\nI am inside DisplayArray\n";

cout<<"tempArray is pointing to location:\t"<<tempArray<<endl;

cout<<"Address of tempArray:\t"<<&tempArray<<endl;

cout<<"\n\n\nData in Array:"<<endl;

for(int i = 0; i< size ; i++)

{

cout<<"tempArray["<<i<<"]\t"<<tempArray[i]<<"\tAddress of tempArray["<<i<<"]\t\t"<<&tempArray[i]<<endl;

}

}

void main()

{

int size = 5;

int\* myArray = GetArrayFromUser(size);

if (myArray != 0) //Why do we need this check?

{

UpdateArray(myArray, size);

DisplayArray(myArray, size);

cout << "\n\n\nI am inside Main()\n";

cout << "myArray is pointing to location:\t" << myArray << endl;

cout << "Address of myArray:\t" << &myArray << endl;

delete[] myArray;

}

}

**Part 2**

#include<iostream>

using namespace std;

void SwapByValue(int a, int b)

{

int temp = a;

a = b;

b = temp;

cout<<"\n\n\nCheck values inside SwapByValue:"<<endl;

cout<<"\n\n\na=\t"<<a<<endl;

cout<<"b=\t"<<b<<endl;

cout<<"\n\n\nCheck Address of Variables inside SwapByValue:"<<endl;

cout<<"Address of a:\t"<<&a<<endl;

cout<<"Address of b:\t"<<&b<<endl;

}

void SwapByReference(int& a, int& b)

{

int temp = a;

a = b;

b = temp;

cout<<"\n\n\nCheck Address of Variables inside SwapByReference:"<<endl;

cout<<"Address of a:\t"<<&a<<endl;

cout<<"Address of b:\t"<<&b<<endl;

}

void main()

{

int x = 5;

int y = 10;

cout<<"Original Values:"<<endl;

cout<<"\n\n\nx = "<<x<<endl;

cout<<"y = "<<y<<endl;

cout<<"\n\n\nCheck Addresses of variables in main():"<<endl;

cout<<"\n\n\nAddress of x\t"<<&x<<endl;

cout<<"Address of y\t"<<&y<<endl;

SwapByValue(x, y);

cout<<"\n\n\nValues in Main() after SwapByValue:"<<endl;

cout<<"x = "<<x<<endl;

cout<<"y = "<<y<<endl;

SwapByReference(x, y);

cout<<"\n\n\nValues in Main() after SwapByReference:"<<endl;

cout<<"x = "<<x<<endl;

cout<<"y = "<<y<<endl;

}

**Part 3** #include<iostream>

using namespace std;

void SwapByPointers(int\* a, int\* b) //Copies of pointers will be generated in activation record of SwapByPointers

{

cout<<"\n\n\nValues in SwapByPointer():"<<endl;

cout<<"a = "<<a<<endl;

cout<<"b = "<<b<<endl;

cout<<"Address of a = "<<&a<<endl;

cout<<"Address of b = "<<&b<<endl;

int\* temp = a;

a = b;

b = temp;

}

void SwapByPointersAgain(int\* a, int\* b) //Copies of pointers will be generated in activation record of SwapByPointersAgain

{

cout<<"\n\n\nValues in SwapByPointerAgain():"<<endl;

cout<<"a = "<<a<<endl;

cout<<"b = "<<b<<endl;

cout<<"Address of a = "<<&a<<endl;

cout<<"Address of b = "<<&b<<endl;

int temp = \*a;

\*a = \*b;

\*b = temp;

}

void main()

{

int x = 5;

int y = 10;

int\* xPtr = &x;

int\* yPtr = &y;

cout<<"Original Values:"<<endl;

cout<<"\n\n\nx = "<<x<<endl;

cout<<"y = "<<y<<endl;

cout<<"\n\n\nCheck Addresses of variables in main():"<<endl;

cout<<"\n\n\nAddress of x\t"<<&x<<endl;

cout<<"Address of y\t"<<&y<<endl;

cout<<"xPtr = "<<xPtr<<endl;

cout<<"yPtr = "<<yPtr<<endl;

cout<<"Address of xPtr = "<<&xPtr<<endl;

cout<<"Address of yPtr = "<<&yPtr<<endl;

SwapByPointers(xPtr, yPtr);

cout<<"\n\n\nValues in Main() after SwapByPointer:"<<endl;

cout<<"x = "<<x<<endl;

cout<<"y = "<<y<<endl;

SwapByPointersAgain(xPtr, yPtr);

cout<<"\n\n\nValues in Main() after SwapByPointerAgain:"<<endl;

cout<<"x = "<<x<<endl;

cout<<"y = "<<y<<endl;

}

**Part 4**

#include<iostream>

using namespace std;

void Swap(int\* & a, int\*& b)

{

cout<<"\n\n\nInside Swap:\n";

cout<<"Address of a:\t"<<&a<<endl;

cout<<"Address of b:\t"<<&b<<endl;

int\* temp = a;

a = b;

b = temp;

}

int\* GetArray()

{

int size = 10;

int\* aPtr = new int[size];

cout<<"\n\n\nInitialize and Print Array:\n";

for(int i = 0 ; i < size ; i++)

{

aPtr[i] = i+1;

cout<<aPtr[i]<<endl;

}

delete[] aPtr;

return aPtr;

}

void main()

{

int x = 5;

int y = 10;

int\* xPtr = &x;

int\* yPtr = &y;

cout<<"Original Values:\n";

cout<<"Address of xPtr:\t"<<&xPtr<<endl;

cout<<"Address of yPtr:\t"<<&yPtr<<endl;

cout<<"\*xPtr:\t"<<\*xPtr<<endl;

cout<<"\*yPtr:\t"<<\*yPtr<<endl;

Swap(xPtr, yPtr);

cout<<"Values after Swap():\n";

cout<<"\*xPtr:\t"<<\*xPtr<<endl;

cout<<"\*yPtr:\t"<<\*yPtr<<endl;

int\* myArray = GetArray(); //What went wrong in this function. Fix this issue.

cout << "I am back in main()" << endl;

cout<<"\n\n\nmyArray[0]:\t"<<myArray[0]<<endl;

cout<<"myArray[5]:\t"<<myArray[5]<<endl;

cout<<"myArray[9]:\t"<<myArray[9]<<endl;

}