/\*Problem 1: | Pointer Arithmetic

Write a program which declares an array of 5 numbers. Initialize the array with 3, 9, 1, 15, 3. Now

use the

• Increment operator

• Decrement operator

in pointers to print the address and values of each index. Also compare all values of pointers

and display if some values are equal.\*/

#include <iostream>

using namespace std;

int main()

{

// For Increment Operator :

cout << "----------------Using Increment operator------------" << endl;

int arr[5] = { 3, 9, 1, 15, 3 };

int\* ptr = arr;

cout << "Address\t\tValue" << endl;

for (int i = 0; i < 5; i++)

{

cout << (ptr + i) << "\t" << \*(ptr + i) << std::endl;

}

cout << "\nComparing values..." << endl;

for (int i = 0; i < 5; i++)

{

for (int j = i + 1; j < 5; j++)

{

if (\*(ptr + i) == \*(ptr + j))

{

cout << "Values at index " << i << " and " << j << " are equal" << endl;

}

}

}

// For Decrement Operator :

cout <<endl <<"----------------Using Decrement operator------------" << endl;

cout << "Address\tValue\n";

for (int i = 5; i >=0 ; i--)

{

cout << ptr << "\t" << \*ptr << endl;

ptr--;

}

cout << endl << "Comparing values: " << endl;

for (int i = 0; i >=5 ; i++)

{

for (int j = i + 1; j < 5; j++)

{

if (\*(arr + i) == \*(arr + j))

{

cout << "Values at index " << i << " and " << j << " are equal." << endl;

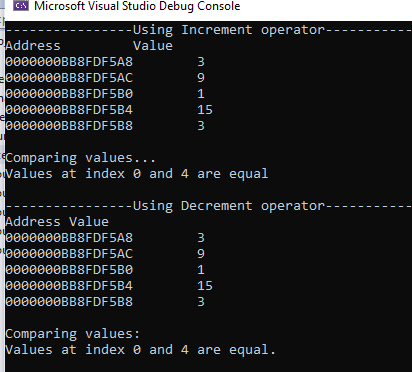
}

}

}

return 0;

}



/\*Problem 2: | Pointer with function

Write a program for passing pointer to a function primaryCheck() while returns true if number is

prime otherwise return false.\*/

#include <iostream>

#include <cmath>

using namespace std;

bool primary\_check(int \*ptr);

int main()

{

int num;

cout << "Enter a number: ";

cin >> num;

int\* ptr;

\*ptr = num;

primary\_check(ptr);

bool type=primary\_check(ptr);

if (type == true)

{

cout << "Number is prime";

}

else

{

cout << "Not a Prime Number";

}

return 0;

}

bool primary\_check(int \*ptr)

{

if (\*ptr <= 1)

{

return false;

}

for (int i = 2; i <= sqrt(\*ptr); i++)

{

if (\*ptr % i == 0)

{

return false;

}

}

return true;

}

/\*Problem 3: | Double pointers\*/

#include <iostream>

using namespace std;

int main()

{

int firstvalue = 15, secondvalue = 25;

cout << "----------------Values Before Pointers--------------" << endl;

cout << "First Value : " << firstvalue << endl;

cout << "second Value : " << firstvalue << endl;

int\* p1, \* p2, \*\* p3, \*\* p4;

// p1 = address of firstvalue

p1 = &firstvalue;

// p2 = address of secondvalue

p2 = &secondvalue;

// p3 = address of firstpointer

p3 = &p1;

// p4 = address of secondpointer

p4 = &p2;

// value pointed by p1 = 100

\*p1 = 100;

// value pointed by p2 = value pointed by p1

\*p2 = \*p1;

// p1 = p2 (address of pointer is copied or not)

p1 = p2;

// p3 = p4 (check address of pointer is copied or not)

p3 = p4;

// value pointed by p1 = 200

\*p1 = 200;

cout << endl << "----------------------After Using Pointers----------------" << endl;

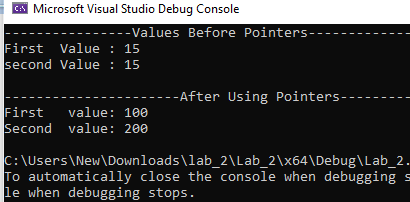
// print firstvalue, secondvalue

cout << "First value: " << firstvalue << endl;

cout << "Second value: " << secondvalue << endl;

return 0;

}



/\*Problem 6: | Pointers\*/

#include <iostream>

using namespace std;

void print\_array(int arr[5])

{

int\* ptr;

ptr = arr;

for (int i = 0; i < 5; i++)

{

cout << "Value at index " << i << " is :" << \*ptr;

ptr++;

cout << endl;

}

}

int main()

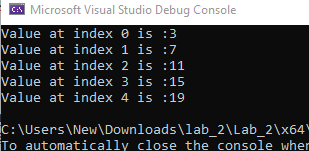
{

int arr[5] = { 3,7,11,15,19 };

print\_array(arr);

return 0;

}



/\*Problem 5: | pointers with function

Write a C++ program to display a matrix of size 3x3.

Passing double pointer to a function print() and display matrix.\*/

#include <iostream>

using namespace std;

void print(int\* ptr, int row, int col);

int main()

{

//suppose 3x3 matrix is:

int matrix[3][3] = { {1, 2, 3}, {4, 5, 6}, {7, 8, 9} };

cout << "Matrix: " << endl;

print(\*matrix, 3, 3);

return 0;

}

void print(int\* ptr, int row, int col)

{

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

{

for (int j = 0; j < col; j++)

{

cout << \*(ptr + i \* col + j) << " ";

}

cout << endl;

}

}

