/\*Problem 1: | Pointer with 2d array:

Write a code to input a 2 by 2 matrix using pointers.

Then add value 100 in all values and print it in matrix form.\*/

#include <iostream>

using namespace std;

void inputMatrix(int\* matrix, int rows, int columns)

{

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

{

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

{

cin >> \*(matrix + i \* columns + j);

}

}

}

void addValue(int\* matrix, int rows, int columns, int value)

{

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

{

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

{

\*(matrix + i \* columns + j) += value;

}

}

}

void printMatrix(int\* matrix, int rows, int columns)

{

for (int i = 0; i < rows; i++) {

for (int j = 0; j < columns; j++) {

cout << \*(matrix + i \* columns + j) << " ";

}

cout << endl;

}

}

int main()

{

const int rows = 2, columns = 2;

int matrix[rows][columns];

cout << "Enter the elements of the 2x2 matrix: " << endl;

inputMatrix(\*matrix, rows, columns);

cout << "Matrix before adding 100: " << endl;

printMatrix(\*matrix, rows, columns);

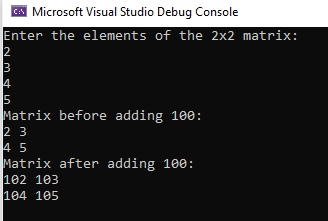
addValue(\*matrix, rows, columns, 100);

cout << "Matrix after adding 100: " << endl;

printMatrix(\*matrix, rows, columns);

return 0;

}



/\* Problem 2: | Dynamic Memory Allocation

Define an int\* pointer variable a. Then:

1. Use new to make a point to a dynamic array of 5 cells of type int.

2. Write a loop to fill a with values 3, 7, 11, 15, 19.

3. Using Print function to print values stored in a.\*/

#include <iostream>

using namespace std;

void arr(int\* a, int size)

{

cout << "------Values Stored In Array are------------" << endl;

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

{

cout << a[i] << " ";

}

}

int main()

{

const int size = 5;

int\* a= new int[size];

cout << "-------------Enter Five Nubers For Array----------------" << endl;

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

{

cout << "Enter Number of array[ "<<i<<" ] : ";

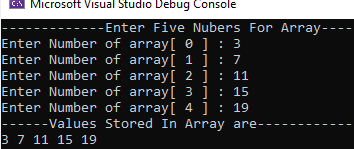
cin >> a[i];

}

arr(a, size);

return 0;

}



/\*Problem 3: | Dynamic Memory Allocation\*/

#include <iostream>

using namespace std;

int function(int\* ptr, int size)

{

ptr = ptr - size;

cout << endl;

cout << "The Duplicate Array:" << endl;

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

{

cout<< \*ptr<<" ";

\*ptr++;

}

return \*ptr;

}

int main()

{

int\* ptr,size;

cout << "Enter the size of an array you want to print : "<<endl;

cin >> size;

ptr = new int[size];

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

{

cout << "Enter Number of Array[ " << i << " ] "<<endl<<"\t";

cin >> \*ptr;

\*ptr++;

}

ptr = ptr - size;

cout << endl << "----------------------" << endl;

cout << "The orginial Array" <<endl;

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

{

cout << \*ptr<<" ";

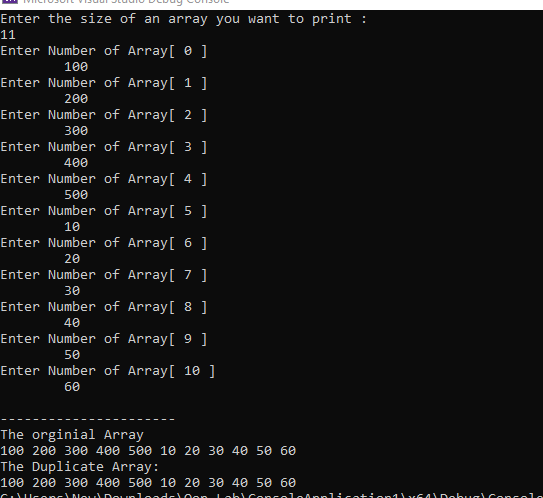
\*ptr++;

}

function(ptr, size);

return 0;

}



/\*Problem 5: (Double pointers, Dynamic Memory Allocation)\*/

#include <iostream>

using namespace std;

void fillElements(int\*\*& matrix, int rows)

{

int columns;

matrix = new int\* [rows];

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

{

cout << "Enter the number of columns in row " << i + 1 << ": ";

cin >> columns;

matrix[i] = new int[columns];

cout << "Enter the elements in row " << i + 1 << ": ";

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

{

cin >> matrix[i][j];

}

}

}

void printMatrix(int\*\* matrix, int rows)

{

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

{

for (int j = 0; j < (sizeof(matrix[i]) / sizeof(matrix[i][0])); j++)

{

cout << matrix[i][j] << " ";

}

cout <<endl;

}

}

int main()

{

cout << "Hello world";

cout << endl;

int\*\* matrix;

int rows;

cout << "Enter the number of rows: ";

cin >> rows;

fillElements(matrix, rows);

cout << "The matrix is:" << endl;

printMatrix(matrix, rows);

return 0;

}

