Problem 5:

#include<iostream>

using namespace std;

int main()

{

/\*int size;

cin >> size;\*/

int\* ptr = new int[999], n;

for (int i = 0; i < 999; i++)

{

cout << "Enter Array: ";

cin >> n;

if (n == -1)

{

exit(0);

int\* ptr1 = new int[i];

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

{

ptr1[i] = ptr[i];

}

}

else

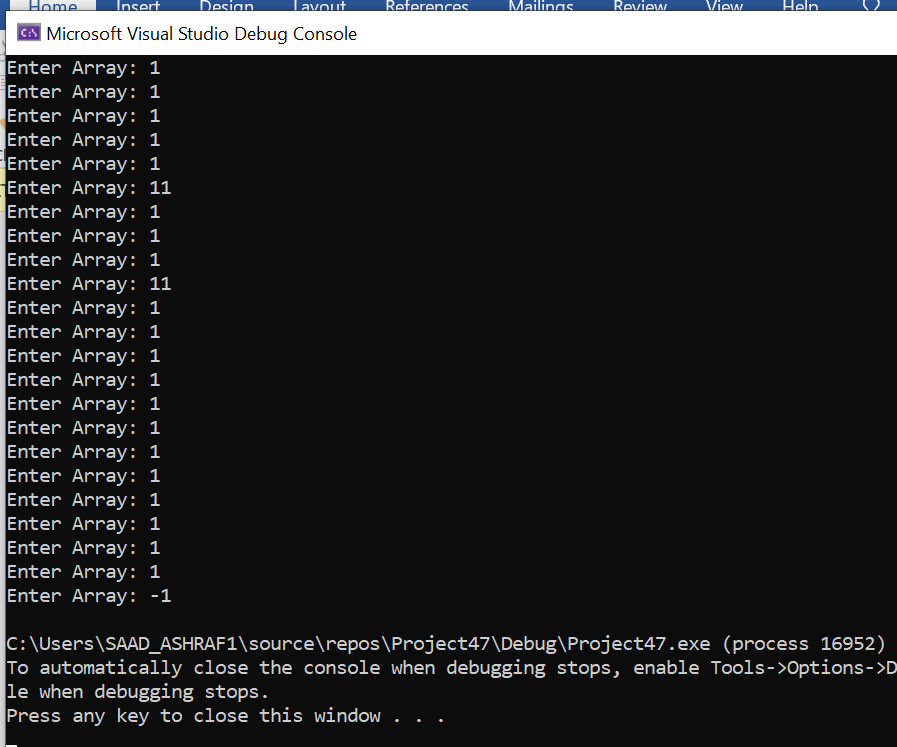
{

ptr[i] = n;

}

}

}



Problem 6:

#include <iostream>

using namespace std;

void input(int\*\* a, int\*\* b, int r1, int c1, int r2, int c2) {

cout << endl << "Enter elements of matrix 1:" << endl;

for (int i = 0; i < r1; ++i)

for (int j = 0; j < c1; ++j)

{

cout << "Enter element a" << i + 1 << j + 1 << " : ";

cin >> a[i][j];

}

// Storing elements of second matrix.

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

for (int i = 0; i < r2; ++i)

for (int j = 0; j < c2; ++j)

{

cout << "Enter element b" << i + 1 << j + 1 << " : ";

cin >> b[i][j];

}

}

void Mulit(int\*\* a, int\*\* b, int\*\* mult, int r1, int c1, int r2, int c2) {

for (int i = 0; i < r1; ++i)

for (int j = 0; j < c2; ++j)

{

mult[i][j] = 0;

}

for (int i = 0; i < r1; ++i)

for (int j = 0; j < c2; ++j)

for (int k = 0; k < c1; ++k)

{

mult[i][j] += a[i][k] \* b[k][j];

}

}

void Out\_Put(int\*\* mult, int r1, int c2) {

cout << endl << "Output Matrix: " << endl;

for (int i = 0; i < r1; ++i)

for (int j = 0; j < c2; ++j)

{

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

if (j == c2 - 1)

cout << endl;

}

}

int main()

{

int r1, c1, r2, c2, i, j, k;

cout << "Enter rows and columns for first matrix: ";

cin >> r1 >> c1;

cout << "Enter rows and columns for second matrix: ";

cin >> r2 >> c2;

// If column of first matrix in not equal to row of second matrix,

// ask the user to enter the size of matrix again.

while (c1 != r2)

{

cout << "Error! column of first matrix not equal to row of second.";

cout << "Enter rows and columns for first matrix: ";

cin >> r1 >> c1;

cout << "Enter rows and columns for second matrix: ";

cin >> r2 >> c2;

}

int\*\* a = new int\* [r1];

int\*\* b = new int\* [r2];

int\*\* mult = new int\* [r1];

for (i = 0; i < r1; i++) {

a[i] = new int[c1];

}

for (i = 0; i < r2; i++) {

b[i] = new int[c2];

}

for (i = 0; i < r1; i++) {

mult[i] = new int[c1];

}

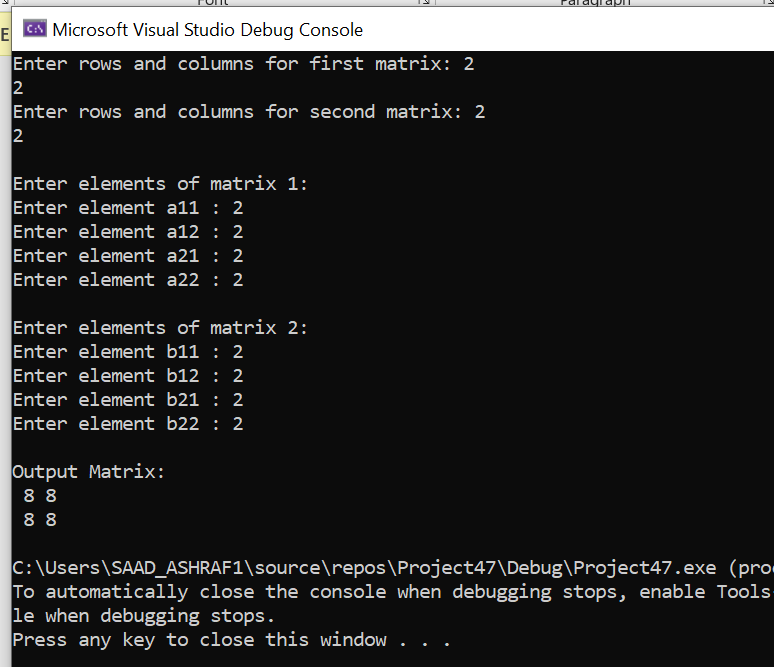
input(a, b, r1, c1, r2, c2);

Mulit(a, b, mult, r1, c1, r2, c2);

Out\_Put(mult, r1, c2);

return 0;

}



Problem 7:

#include <iostream>

#include <string>

using namespace std;

int main(){

int rows;

cout << "Enter rows of array: ";

cin >> rows;

int\* numbers = new int[rows];

//declaration of array

int\*\* array = new int\* [rows]; //jagged array

int i = 0;

while ( i < rows)

{

cout << "Enter column in row " << i << ": ";

cin >> numbers[i];

array[i] = new int[numbers[i]]; //creating new dynamic memory

i++;

}

//Input values in array

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

{

for (int j = 0; j < numbers[i]; j++)

{

cout << "Row " << i << ":- input value " << i \* numbers[i] + j << ": ";

cin >> array[i][j];

}

}

//Output values of array

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

{

for (int j = 0; j < numbers[i]; j++)

{

cout << array[i][j] << " "; //printing the jagged array

}

cout <<endl;

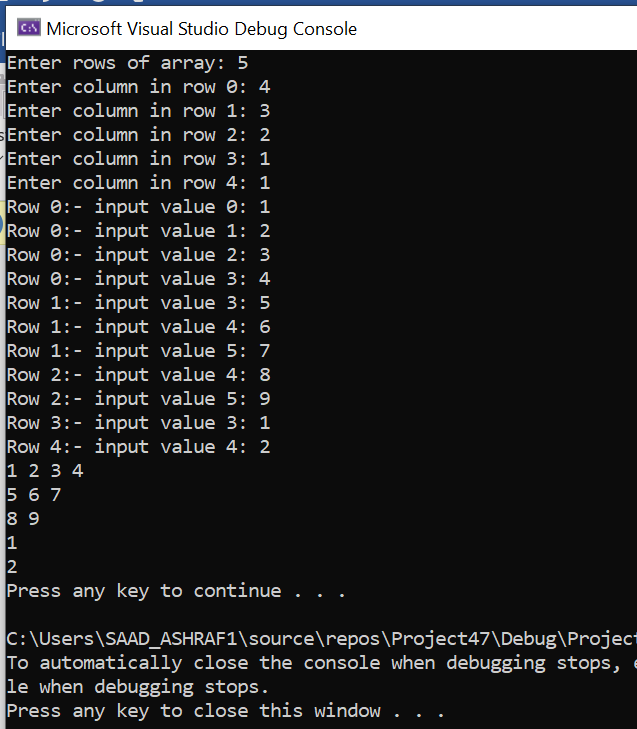
}

delete array; //deleting dynamic array

system("pause");

return 0;

}



Task 8: #include<iostream>

#include<string>

using namespace std;

int main() {

int s;

string a;

cout << "Enter Size of Array : " << endl;;

cin >> s;

char\*\* arr;

cout << endl;

int i = 0;

arr = new char\* [s];

int size1 = 0;

while (true) {

cin >> a;

size1 = a.length();

arr[i] = new char[size1 + 1];

int j = 0;

for (j = 0; a[j] != '\0'; j++) {

arr[i][j] = a[j];

}

arr[i][j + 1] = NULL;

if (i == s - 1) {

break;

}

i++;

}

cout << endl;

i = 0;

while (i < s) {

cout << endl;

int k = 0;

while (k < strlen(arr[i]) - 1)

{

cout << arr[i][k] << "|";

k++;

}

cout << "NULL" << "|";

cout << endl;

i++;

}

}

