Task 1;

#include<iostream>

using namespace std;

int main()

{

int a[2][2] = {0};

cout << "enter values in array\n";

int (\*ptr)[2] = a;

for (int i = 0; i < 2; i++)

{

for (int j = 0; j < 2; j++)

{

cin >> \*(\*(ptr+i)+j);

}

}

for (int i = 0; i < 2; i++)

{

for (int j = 0; j < 2; j++)

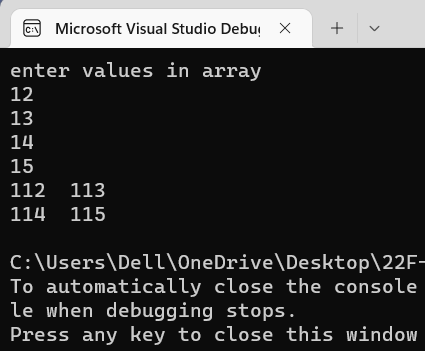
{

cout << \*(\*(ptr + i) + j)+100<<" ";

}cout << endl;

}

}



Task 2:

#include<iostream>

using namespace std;

void print(int a[5])

{

cout << "array is ";

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

{

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

}

}

int main()

{

int\* a, arr[5];

int\* ptr = new int[5];

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

{

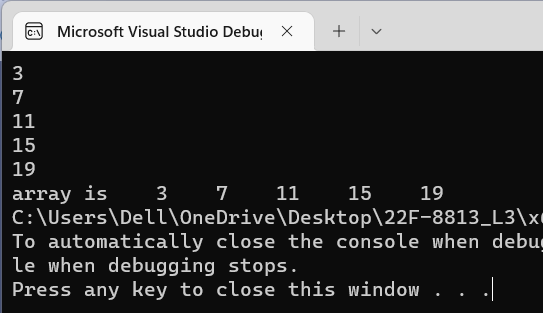
cin >> ptr[i];

}

print(ptr);

delete []ptr;

}



Task 3:

#include<iostream>

using namespace std;

int \*copyfun(int a[4], int s)

{

int copy[4] = {0};

int\* ptr = new int[2];

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

{

ptr[i] = a[i];

copy[i] = ptr[i];

}cout << "copied array is \n";

for (int i = 0; i < 4; i++)

{

cout << copy[i];

}

return ptr;

}

int main()

{

int size;

int a[4] = { 1,2,3,4};

size = sizeof(a) / sizeof(a[0]);

cout << copyfun(a, size);

cout << endl;

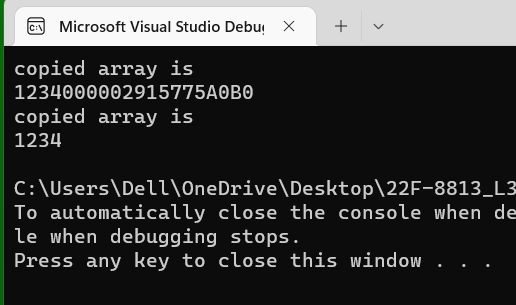
int\* ptr = copyfun(a, size);

cout << endl;

delete ptr;

ptr = NULL;

}



Task 4:

#include<iostream>

using namespace std;

void Sum(int\*\* p, int row, int col, int\*\* q, int row2, int col2) // this function will take sum of two matricesand return the resultant matrix.

{

int c[2][2] = { 0 };

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

{

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

{

c[i][j]=p[i][j] + q[i][j];

}

}

for (int i = 0; i < 2; i++)

{

for (int j = 0; j < 2; j++)

{

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

}

cout << endl;

}

}

void Display(int\*\* p, int row, int col)// this function will output the p matrix to console

{

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

{

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

{

cout << p[i][j];

}

}

int row2 = 2, col2 = 2;

int a[2][2] = { {1,2},{3,4} };

int\*\* q = new int\*[2] ;

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

{

q[i] = new int[2];

}

for (int i = 0; i < 2; i++)

{

for (int j = 0; j < 2; j++)

{

q[i][j] = i;

}

}

Sum( p, row, col,q, row2, col2);

}

void input(int\*\* p, int row, int col)// this function will input the p matrix

{

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

{

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

{

p[i] = new int[col];

}

}

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

{

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

{

cin >> p[i][j];

}

}

Display(p, row, col);

}

int main()

{

int row=0,col=0;

cout << "enter rows\n";

cin >> row;

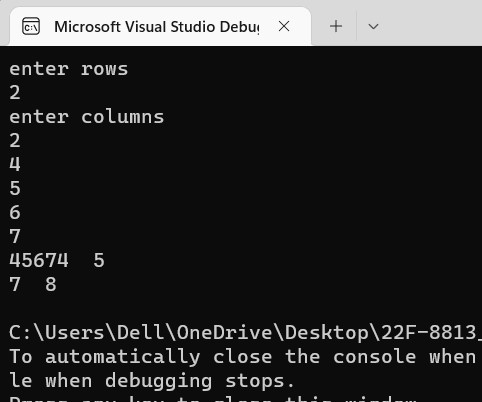
cout << "enter columns\n";

cin >> col;

int\*\* ptr = new int\* [row];

input(ptr, row, col);

}



Task 5:

#include<iostream>

using namespace std;

void print(int\*\* pt, int row, int c)

{

cout << "array is \n";

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

{

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

{

cout << pt[i][j];

}cout << endl;

}

}

void fill(int\*\* p, int r)

{

int col = 5;

for (int count = 0; count < r; ++count)

{

cout << "enter col of row " << count<<endl;;

cin >> col;

p[count] = new int[col];

cout << "enter col entities \n";

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

{

cin >> p[count][i];

}

}

print(p, r, col);

}

int main()

{

int row = 0, col = 0;

cout << "enter row\n";

cin >> row;

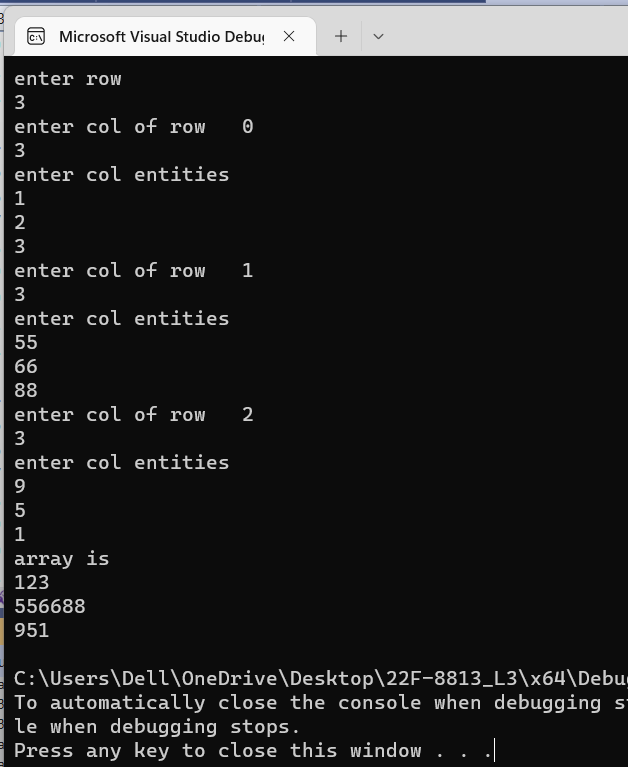
int\*\* ptr = new int\* [3];

fill(ptr,row);

delete[]ptr;

ptr = NULL;

}



Task 6:

#include<iostream>

#include<string>

using namespace std;

int main()

{

int count=0,c=0;

string str;

getline(cin, str);

char ch[100][100];

int a[10]={0};

for (int i = 0; i < str.length(); i++)

{

if (str[i] == '32' || str[i] == '.') {

count++;

}

}

char \*\* ptr = new char\* [count];

for (int i = c; i < count; i++)

{

if (str[i] != '32' || str[i] != '.')

{

c++;

}

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

{

ptr[count] = new char[c];

ch[i][j] = str[i];

}

}

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

{

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

{

cout << ch[i][j];

}

}

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

{

delete ptr[i];

}delete[]ptr;

}