/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*TOP DOCUMENTATION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*

\* (1) *Add your Top Documentation in this section*

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream> //\_\_(2)\_\_ to include the iostream library

#include <string>

using namespace std; //std is the abbreviation for \_\_(3)\_\_

void swapping(string &a, string &b) //swap the content of \_\_(4)\_\_ and \_\_(5)\_\_ that are type \_\_(6)\_\_

{

string temp;

temp = a;

a = b;

b = temp;

}

void swapping(int& a, int& b) //swap the content of \_\_(7)\_\_ and \_\_(8)\_\_ that are type \_\_(9)\_\_

{

int temp;

temp = a;

a = b;

b = temp;

}

void display(string arraySt[], int arrayIn[][10], int rsize, int csize) //\_\_(10)\_\_ the arrays

{

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

cout << arraySt[i] << " ";

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

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

cout << endl;

}

cout << endl;

}

void bubbleSort(string arraySt[], int arrayIn[][10], int rsize, int csize) //sorts by the \_\_(11)\_\_ array

{

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

{

int swaps = 0; // swap is used as a \_\_(12)\_\_ to detect any swap has happened

for (int j = 0; j < rsize - i - 1; j++)

{

if (arraySt[j] > arraySt[j + 1]) //when the current item is \_\_(13)\_\_ than the next... SWAP

{

swapping(arraySt[j], arraySt[j + 1]);

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

swapping(arrayIn[j][k], arrayIn[j + 1][k]);

swaps = 1; //set swap flag

}

}

if (!swaps)

break; // \_\_(14)\_\_ swap in this pass, so array is sorted

}

}

int main()

{

int rows, columns; // the \_\_(15)\_\_ number of rows and columns in the arrays

string arrSt[10]; //create a string array no greater than \_\_(16)\_\_ elements

int arrIn[10][10]; //create a \_\_(17)\_\_ dimensional integer array no greater than \_\_(18)\_\_ elements

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

cin >> rows;

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

cin >> columns;

if (rows > 10 || columns > 10) // \_\_(19)\_\_check for row and \_\_(20)\_\_

cout << "There is a limit of 10 elements for rows and columns" << endl;

else

{

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

{

cout << "Enter string element number " << i + 1 << ": ";

cin >> arrSt[i];

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

{

cout << "Enter integer element number [" << i + 1 << ", " << j + 1 << "] : ";

cin >> arrIn[i][j];

}

}

cout << "Array before Sorting: " << endl;

display(arrSt, arrIn, rows, columns); //displays the \_\_(21)\_\_list

bubbleSort(arrSt, arrIn, rows, columns); //\_\_(22)\_\_ the data by the \_\_(23)\_\_ array

cout << "Array after Sorting: " << endl;

display(arrSt, arrIn, rows, columns); //displays the \_\_(24)\_\_ list

}

cout << endl;

system("pause");

return 0;

}

//Function swapping defined with different types of parameters is known as \_\_(25)\_\_.