SALİH EREN DEĞİRMENCİ

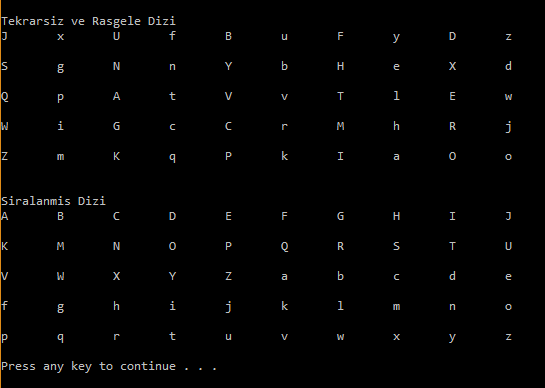
G181210017

SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

PROGRAMLAMAYA GİRİŞ DERSİ D GRUBU

2. ÖDEV



The output want from us is given above.I need a 5x10 char array filled with letters.Letters must arrayed be one uppercase an one lowercase.This means if column numbers remaining 0 by divided 2,

It will be a uppercase letter.Than there is an another rule that if a letter used once , it weill never be used again so ı need check it if it is used, than replace it with a non used letter.Then pring the non sorted matix.After that finally sort it by alphabetically (first uppercase letters,than lowercase latter.).According to ascii table ‘A’ letters code is 65 and it increasing to one by one to ‘Z’.as same as that letter ‘a’ starts with 97 and it increasing one by one to ‘z’.I used sort algorithm. Sort algorithm sort the letters according to their code numbers.It puts to minimum code number to [0][0]’th element of matrix and highest code number to [5][10]’th element of matrix..finally print the sorted matrix.

MY CODES

#include "iostream" //in-out library

#include "time.h"//for random function

#include "algorithm"//for sorting algorithm

using namespace std;

int main() {

srand(time(NULL)); // parameter of seed

int randletter; //This a integer need to declare a random letter.

int i, j; //for getting every row and column one by one.

char matrixelements; //elements (letters) of matrix.

const int row = 5;//row range.

const int col = 10;//column range.

char matrix[row][col];//Matrix.

for (i = 0; i < row; i++)//For every row 0-4.

{

for (j = 0; j < col; j++)//For every column 0-9.

{

randletter = rand() % 26;//generates a random integer.

if (j % 2 == 1) // if column number is odd,It will be lowercase letter.

{

matrixelements = 'a' + randletter; //adding the integer it created to decide a letter a-z.

}

else // if column number is even,It will be uppercase letter.

{

matrixelements = 'A' + randletter; //adding the integer it created to decide a letter A-Z.

}

for (int controlrow = 0; controlrow < row; controlrow++) //Creates another matrices row for controlling if there is repated letter.

{

for (int controlcol = 0; controlcol < col; controlcol++) //Creates another matrices column for controlling if there is repated letter.

{

if (matrix[controlrow][controlcol] == matrixelements)//If there is a same letter in matrix :

{

if (j % 2 == 0 )//If column number is an even number replace it with a lowercase letter.

{

randletter = rand() % 26;

matrixelements = 'a' + randletter;

}

else {

randletter = rand() % 26;//If column number is an odd number replace it with a uppercase letter.

matrixelements = 'A' + randletter;

}

controlrow = 0;

controlcol = 0;

}

}

}

matrix[i][j] = { matrixelements };// non arrayed non repated matrix.

}

}

cout << "\*\*\*\*\*\*\*\*\*\* Non Arrayed Matrix \*\*\*\*\*\*\*\*\*\*\* " << endl;

for (int i = 0; i < row; i++)//printing the non arrayed non repated matrix.

{

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

{

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

}

cout << endl;

}

sort(&matrix[0][0], &matrix[0][0] + 5 \* 10);//Sorting matrix according to the alphabet.

cout << "\*\*\*\*\*\*\*\*\*\* Arrayed Matrix \*\*\*\*\*\*\*\*\*\*\* " << endl;

for (int i = 0; i < row; i++)//printing the arrayed non repated matrix.

{

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

{

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

}

cout << endl;

}

system("pause");

return 0;

}