Створити функцію для знаходження значення, яке знаходиться в межах деякого діапазону (аналогічно до шифру Цезаря) після додавання до нього заданого числа (ціле число або символ).

// ConsoleApplication21.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <iostream>

using namespace std;

template <class T1, class T2>

T1 getValue(T1 minValue, T1 maxValue, T2 shift, T1 value)

{

return (value - minValue + shift) % (maxValue - minValue + 1) + minValue;

}

int main()

{

//---------------- min=1, max=8, shift =500

int v1 = 9;

v1 = getValue(1,8,500,v1);

printf("v1=%d", v1);

//----------------

char c = 'B';

c = getValue('A', 'Z', 500, c);

printf("c=%c",c);

system("pause");

return 0;

}

///--------------------------------------------------------------------

Задача 2. Нехай , де . Визначити .

// ConsoleApplication21.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <iostream>

using namespace std;

double x(int i)

{

if (i < 2)

return 1;

else

return x(i - 1) + 2 \* x(i - 2);

}

double x\_(int n)

{

if (n < 2)

return 1;

else

{

double xi\_1=1, xi\_2=1, xi;

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

{

xi = xi\_1 + 2 \* xi\_2; //Згідно з формулою

xi\_2 = xi\_1;

xi\_1 = xi;

}

return xi;

}

}

int main()

{

int n;

printf("n=");

cin >> n;

double res = x(n);

printf("Result=%f\n",res);

//--------------------

double res1 = x\_(n);

printf("Result=%f\n", res1);

system("pause");

return 0;

}

Задача 3. Тренажер «Перекладач». Два масиви : в одному зберігаються слова на українській мові, в іншому під тим же номером зберігається переклад на англійській. Випадковим чином вибравши слово на українській мові, запропонувати користувачу ввести переклад і перевірити правильність. Таким чином перебрати усі слова випадковим чином до тих пір, поки не буде переглянуто усі слова. В кінці вивсти слова, у яких користувач робив помилки.

// ConsoleApplication21.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <iostream>

#include <string>

using namespace std;

int getNextIndex(int\* control, int length)

{

int index;

do

{

index = rand() % length;

} while (control[index]!=1); //Генеруємо індекст до тих пір, поки не знайдемо елемент масиву, який дорівнює 1

return index;

}

int\* TestWords(string ukrWords[], string engWords[], int length)

{

int \* control = new int[length];

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

{

control[i] = 1; //Якщо 1, то слово з цим номером ще не розглянуто

}

//-----------

int checkedWordsCount = 0;

while (checkedWordsCount<length)

{

int index = getNextIndex(control, length);

printf("----------------------------\n");

printf("Ukrainian : %s\n",ukrWords[index].data());

printf("English :");

string wordTranslation;

cin >> wordTranslation;

if (wordTranslation == engWords[index])

{

control[index] = 2;

}

else

{

control[index] = 0;

}

checkedWordsCount++;

}

return control;

}

int main()

{

setlocale(LC\_ALL, "Russian");

string ukrWords[5]{ "ручка", "книга", "вiкно", "телефон", "будинок"};

string engWords[5]{ "pen", "book", "window", "phone", "house" };

int \* testResult = TestWords(ukrWords, engWords, 5);

bool errors = false;

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

{

if (testResult[i]==0)

{

printf("%s\n",engWords[i].data());

errors = true;

}

}

if (errors==false)

{

printf("No errors\n");

}

system("pause");

return 0;

}

//===============================

// ConsoleApplication21.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <iostream>

#include <string>

#include <ctime>

using namespace std;

int getNextIndex(int\* wordsNumbers, int& wordsCount)

{

int index = rand() % wordsCount;

int temp = wordsNumbers[index];

wordsNumbers[index] = wordsNumbers[wordsCount - 1];

wordsCount--;

return temp;

}

void TestWords(string ukrWords[], string engWords[], int length, int\*& problemWords, int& problemWordsCount)

{

problemWords = new int[length];

problemWordsCount = 0;

int\* wordsNumbers = new int[length];

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

{

wordsNumbers[i] = i; //Масив містить номери слів, які ще не розглянуто

}

int wordsCount = length;

//-----------

while (wordsCount>0)

{

int index = getNextIndex(wordsNumbers, wordsCount);

printf("----------------------------\n");

printf("Ukrainian : %s\n",ukrWords[index].data());

printf("English : ");

string wordTranslation;

cin >> wordTranslation;

if (wordTranslation != engWords[index])

{

problemWords[problemWordsCount]=index;

problemWordsCount++;

}

}

}

int main()

{

setlocale(LC\_ALL, "Russian");

srand(time(NULL));

string ukrWords[5]{ "ручка", "книга", "вiкно", "телефон", "будинок"};

string engWords[5]{ "pen", "book", "window", "phone", "house" };

int \* problemsWords;

int problemsWordsCount;

TestWords(ukrWords, engWords, 5, problemsWords, problemsWordsCount);

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

{

printf("%s\n",engWords[ problemsWords[i] ].data());

}

if (problemsWordsCount==0)

{

printf("No errors\n");

}

system("pause");

return 0;

}

//=========================================

**Дано двовимірний масив, який містить інформацію про кількість відвідувань фітнес-клубу за K тижнів. Необхідно знайти тиждень, у якому кількість відвідувань була найбільшою (найменшою). Визначати тиждень, протягом якого не було дня, у який кількість відвідувачів була більшою (меншою) за N.**

// ConsoleApplication21.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <iostream>

#include <string>

#include <ctime>

using namespace std;

int\*\* inputArrayFromKeyBoard(int rowsCount, int columnsCount)

{

int\*\* arr = new int\*[rowsCount];

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

{

arr[i] = new int[columnsCount];

}

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

{

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

{

printf("row =%d, column =%d :",i,j);

cin >> arr[i][j];

}

}

return arr;

}

int\*\* generateArray(int rowsCount, int columnsCount, int minValue, int maxValue)

{

int\*\* arr = new int\*[rowsCount];

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

{

arr[i] = new int[columnsCount];

}

int rangeLength = maxValue - minValue;

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

{

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

{

arr[i][j]=minValue+rand()%rangeLength;

}

}

return arr;

}

//------------------------------------------

int getMAxWeekNumber(int\*\* weeksClientsNumbers, int weekCount)

{

int maxClientsNumber = -1;

int maxWeekNumber = -1;

for (int week = 0; week < weekCount; week++)

{

int sum = 0;

for (int j = 0; j < 7; j++)

{

sum += weeksClientsNumbers[week][j];

}

if (sum>maxClientsNumber)

{

maxClientsNumber = sum;

maxWeekNumber = week;

}

}

return maxWeekNumber;

}

int main()

{

setlocale(LC\_ALL, "Russian");

srand(time(NULL));

int weekCount;

printf("Week count=");

cin >> weekCount;

int\*\* week = generateArray(weekCount, 7, 0, 100);

int maxWeekNumber = getMAxWeekNumber(week, weekCount);

printf("maxWeekNumber =%d", maxWeekNumber);

system("pause");

return 0;

}