Формула для знаходження значення, яке змінюється від min до max, якщо до нього додати shift

// (value-min+shift)%(max-min+1)+min

Шифр Цезаря

char c;

cin >> c;

int shift;

cin >> shift;

char newC = (c - 'A' + shift) % 26 + 'A';

printf("New c = %c");

1. Дано одновимірний масив. Використовуючи покажчики на функції створити функцію, що володіла б наступними функціональними можливостями при перегляді трійок елементів (a[0], a[1], a[2] далі a[3], a[4], a[5], і т.д.):

1) виводила найбільший елемент;

2) виводила найменший елемент;

3) виводила суму елементів.

// ConsoleApplication63.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include<iostream>

using namespace std;

//------- Описуємо правила ----------

double getMax(double a, double b, double c)

{

if (a >= b && a >= c)

return a;

if (b >= c)

return b;

return c;

}

double getMin(double a, double b, double c)

{

if (a <= b && a <= c)

return a;

if (b <= c)

return b;

return c;

}

double getSum(double a, double b, double c)

{

return a+b+c;

}

//-------- Пишемо загальну функцію (де використовуються правила, тобто, у яку передаються правила)

void printArrayByRule(double\* arr, int length, double (\*rule)(double, double, double))

{

for (int i = 2; i < length; i+=3)

{

printf("%.2f\n", rule(arr[i - 2], arr[i - 1], arr[i]) );

}

}

int main()

{

//Дано одновимірний масив.Використовуючи покажчики на функції створити функцію,

//що володіла б наступними функціональними можливостями при перегляді трійок елементів(a[0], a[1], a[2] далі a[3], a[4], a[5], і т.д.) :

// 1) виводила найбільший елемент;

// 2) виводила найменший елемент;

// 3) виводила суму елементів.

double\* a = new double[6]{2,33,45,67,23,9};

int length = 6;

printf("1) max, 2) min 3) sum :");

int answer;

cin >> answer;

double(\*userRule)(double, double, double); //Описано покажчик userRule на функцію

userRule = 0; //Покажчику присвоюємо порожню адресу

switch (answer)

{

case 1:userRule= getMax; //У покажчик userRule записуємо адресу функції getMax

break;

case 2:userRule = getMin; //У покажчик userRule записуємо адресу функції getMin

break;

case 3:userRule = getSum; //У покажчик userRule записуємо адресу функції getSum

break;

}

printArrayByRule(a, length, userRule);

system("pause");

return 0;

}

// ConsoleApplication63.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include<iostream>

using namespace std;

//Дано вимір часу години.хвилини.Підкоректувати цей вимір у разі потреби(наприклад,

// вимір 13.64 треба замінити на 14.04, а вимір 27.25 треба замінити на 03.25).

void correctTime(int& h, int& m) //Випадок, коли окремо години і окремо хвилини

{

h += m / 60;

m %= 60;

h %= 24;

}

//Випадок, коли передається як дійсне число (години –ціл, хв.- дробова частина)

void correctTime(double& time)

{

int h = time;

int m = (time - h) \* 100;

//h += m / 60;

//m %= 60;

//h %= 24;

correctTime(h, m);

time = h + m / 100.0;

}

int main()

{

printf("Hours :");

int hours;

cin >> hours;

printf("Minutes :");

int minutes;

cin >> minutes;

correctTime(hours, minutes);

printf("%d : %d",hours,minutes);

system("pause");

return 0;

}

//Створити функцію, яка для двовимірного масиву mas знаходить суму додатніх елементів

//вказаного рядка rowNumber.

double getSumRow(double\*\* a, int columnCount, int rowIndex)

{

double sum = 0;

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

{

sum += a[rowIndex][j];

}

return sum;

}