Описать процедуру RemoveForInc(A, N), удаляющую из вещественного массива A размера N «лишние» элементы так, чтобы оставшиеся элементы оказались упорядоченными по возрастанию: первый элемент не удаляется, второй элемент удаляется, если он меньше первого, третий — если он меньше предыдущего элемента, оставленного в массиве, и т. д.Например, массив 5.5, 2.5, 4.6, 7.2, 5.8, 9.4 должен быть преобразован к виду 5.5, 7.2, 9.4. Массив A и число N являются входными и выходными параметрами. С помощью этой процедуры преобразовать массивы A, B,С размера NA, NB, NC соответственно и вывести содержимое полученных массивов.

#include<iostream>

using namespace std;

int RemoveForInc(float\* arr, int size) {

int p, i, j;

for (p = 0, i = 1; i < size; i++) {

if (arr[i] < arr[p]) {

for (j = i; j < size - 1; j++)

arr[j] = arr[j + 1];

--size;

i--;

}

else

p++;

}

return size;

}

float randfrom(float min, float max)

{

float range = (max - min);

float div = RAND\_MAX / range;

return round((min + (rand() / div)) \* 10) / 10;

}

void FillArray(float\* const arr, const int size)

{

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

arr[i] = randfrom(1, 15);

}

}

void ShowArray(float\* const arr, const int size)

{

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

cout << arr[i] << " ";

cout << endl;

}

int main()

{

cout<< "enter array size 1 ";

int size1;

cin >> size1;

float \*arr1=new float[size1];

FillArray(arr1, size1);

ShowArray(arr1, size1);

size1 = RemoveForInc(arr1, size1);

for (int i = 0; i < size1; i++)

cout<< arr1[i]<<" ";

cout << endl;

delete[] arr1;

cout << "enter array size 2 ";

int size2;

cin >> size2;

float\* arr2 = new float[size2];

FillArray(arr2, size2);

ShowArray(arr2, size2);

size2 = RemoveForInc(arr2, size2);

for (int i = 0; i < size2; i++)

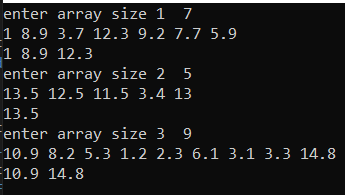
cout << arr2[i] << " ";

cout << endl;

delete[] arr2;

cout << "enter array size 3 ";

int size3;

 cin >> size3;

float\* arr3 = new float[size3];

FillArray(arr3, size3);

ShowArray(arr3, size3);

size3 = RemoveForInc(arr3, size3);

for (int i = 0; i < size3; i++)

cout << arr3[i] << " ";

cout << endl;

delete[] arr3;

return 0;

}

Описать функцию PosLast(S0, S) целого типа, возвращающую номер позиции, начиная с которой в строке S содержится последнее вхождение подстроки S0. Считать, что перекрывающихся вхождений подстрок S0 строка S не содержит. Если в строке S отсутствуют подстроки S0, то функция возвращает 0. Вывести значения этой функции для пяти данных пар строк S0 и S.

#include<iostream>

#include <Windows.h>

#include <string>

using namespace std;

int Finding\_a\_position(string S, string S0)

{

for (int i = size(S) - 1, j, k; i >= 0; i--) {

for (j = size(S0) - 1, k = 0; j >= 0; j--, k++) {

if (S[i - k] != S0[j])

break;

}

if (j == -1)

return i - size(S0) + 1;

else return 0;

}

}

int main()

{

setlocale(LC\_ALL, "ru");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

string S1;

cout << "Введите строку1: ";

cin >> S1;

string S01;

cout << "Введите подстроку1: ";

cin >> S01;

cout << Finding\_a\_position(S1, S01) << endl;;

string S2;

cout << "Введите строку2: ";

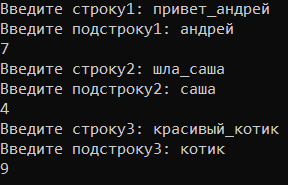
cin >> S2;

string S02;

cout << "Введите подстроку2: ";

cin >> S02;

cout << Finding\_a\_position(S2, S02) << endl;;

 string S3;

cout << "Введите строку3: ";

cin >> S3;

string S03;

cout << "Введите подстроку3: ";

cin >> S03;

cout << Finding\_a\_position(S3, S03) << endl;

}

Описать рекурсивную функцию NOD(A, B) целого типа, находящую наибольший общий делитель (НОД) двух натуральных чисел A и B, используя алгоритм Евклида:

НОД(A, B) = НОД(B, A mod B), если B ≠ 0; НОД(A, 0) = A.

С помощью этой функции найти НОД(A, B), НОД(A, C), НОД(A, D), если даны числа A, B, C, D.

#include<iostream>

using namespace std;

int NOD(int a, int b)

{

return b != 0 ? NOD(b, a % b) : a;

}

int main()

{

setlocale(LC\_ALL, "ru");

int a, b, c, d;

cout << "Введите числа: ";

cin >> a >> b >> c >> d;

cout << "Наименьший общий делитель числел a и b: ";

cout << NOD(a, b) << endl;;

cout << "Наименьший общий делитель числел a и c: ";

cout << NOD(a, c) << endl;

cout << "Наименьший общий делитель числел a и d: ";

cout << NOD(a, d) << endl;

}

