**Лабораторная работа №5**

**«Программирование алгоритмов формирования и обработки списков»**

**Задача:** Вводиться два рандомных списка x и y.

Объедините их в один список z, поместив элементы списка «y» между третьим и четвертым элементами списка «x».

Примечание:

1. Python: Использовать функции и модули.

Модуль должен выполнять функцию генерирования 2-х рандомных списков (x, y) и давать возможность ввода позиции добавления элемента в список, формируя новый список z.

Результат представить:

В виде вывода списка z

1. С++: Использовать функции и DLL

Функция должна выделять динамическую память под объекты списков x, y, z, выводить списки на экран и формировать размер списка z, в конце высвобождая память. Также функция должна содержать в себе DLL библиотеку, которая в свою очередь выполняла бы генерирования рандомных списков (x, y), далее формируя общий список z, после реализовывала возможность добавления позиции элемента в список z.

Результат представить:

В виде вывода списка z

**Алгоритм решения задачи:**



**Рис.1.** Общая блок-схема решения задачи



**Рис.2.** Алгоритм модуля (Module)

**Код программы (Array):**

import Module

N=int(input("Введите количество элементов списка x "))

M=int(input("Введите количество элементов списка y "))

z=Module.fun(N,M)

exit(0)

**Код модуля (Module):**

import random

def fun(N,M):

x=random.sample(range(100),N)

print("\n Список x ",x)

y=random.sample(range(100),M)

print("\n Список y ",y)

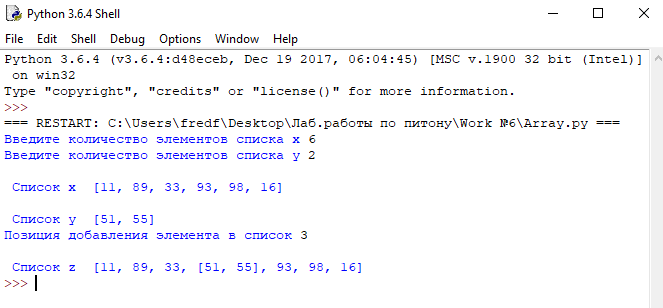
poz=int(input("Позиция добавления элемента в список "))

x.insert(poz,y)

print("\n Список z ",x)

return x

**Результат работы программы:**



**Рис.3.** Результат работы приложения

**C++**

**Код программы (Windows Form):**

#include "iostream"

#include "time.h"

#include "MathFuncsDll.h"

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

int x, y, p;

srand(time(NULL));

x = Convert::ToDouble(textBox1->Text);

y = Convert::ToDouble(textBox2->Text);

p = Convert::ToDouble(textBox3->Text);

outmas(x,y,p);

}

void outmas(int x, int y,int p)

{

int pos, num, num2;

int \*arr = new int[x];

MathFuncs::MyMathFuncs::memory\_arr(num, arr, x);

dataGridView1->ColumnCount = x;

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

{

dataGridView1->Rows[0]->Cells[i]->Value = arr[i];

}

int \*arr2 = new int[y];

MathFuncs::MyMathFuncs::memory\_arr2(num2, arr2, y);

dataGridView2->ColumnCount = y;

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

{

dataGridView2->Rows[0]->Cells[i]->Value = arr2[i];

}

pos = x + y;

int \*c = new int[pos];

MathFuncs::MyMathFuncs::memory\_c(pos,arr,arr2,x,c);

dataGridView3->ColumnCount = pos;

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

{

dataGridView3->Rows[0]->Cells[i]->Value = c[i];

}

int start = p;

int finish = p + y;

int \*z = new int[pos];

MathFuncs::MyMathFuncs::memory\_z(pos,arr,arr2,start,finish,z,p);

dataGridView4->ColumnCount = pos;

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

{

dataGridView4->Rows[0]->Cells[i]->Value = z[i];

}

delete arr;

delete arr2;

delete c;

delete z;

}

**Код заголовочного файла (MathFuncsDll.h):**

#pragma once

#ifdef MATHFUNCSDLL\_EXPORTS

#define MATHFUNCSDLL\_API \_\_declspec(dllexport)

#else

#define MATHFUNCSDLL\_API \_\_declspec(dllimport)

#endif

namespace MathFuncs

{

class MyMathFuncs

{

public:

static MATHFUNCSDLL\_API void memory\_arr(int num, int \*arr, int x);

static MATHFUNCSDLL\_API void memory\_arr2(int num2, int \*arr2, int y);

static MATHFUNCSDLL\_API void memory\_c(int pos, int \*arr ,int \*arr2, int x,int \*c);

static MATHFUNCSDLL\_API void memory\_z(int pos, int \*arr, int \*arr2, int start, int finish, int \*z, int p);

};

}

**Код библиотеки (MathFuncsDll.cpp):**

#include "stdafx.h"

#include "MathFuncsDll.h"

#include <stdexcept>

using namespace std;

namespace MathFuncs

{

void MyMathFuncs::memory\_arr(int num, int \*arr, int x)

{

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

{

num = rand() % 10;

arr[i] = num;

}

}

void MyMathFuncs::memory\_arr2(int num2, int \*arr2, int y)

{

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

{

num2 = rand() % 10;

arr2[i] = num2;

}

}

void MyMathFuncs::memory\_c(int pos, int \*arr, int \*arr2, int x, int \*c)

{

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

{

c[i] = arr[i];

}

for (int i = x; i < pos; i++)

{

c[i] = arr2[i - x];

}

}

void MyMathFuncs::memory\_z(int pos, int \*arr, int \*arr2, int start, int finish, int \*z,int p)

{

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

{

if ((i >= start) && (i < finish))

{

z[i] = arr2[i - start];

}

else if (i < start)

{

z[i] = arr[i];

}

else if (i >= finish)

{

z[i] = arr[i - finish + p];

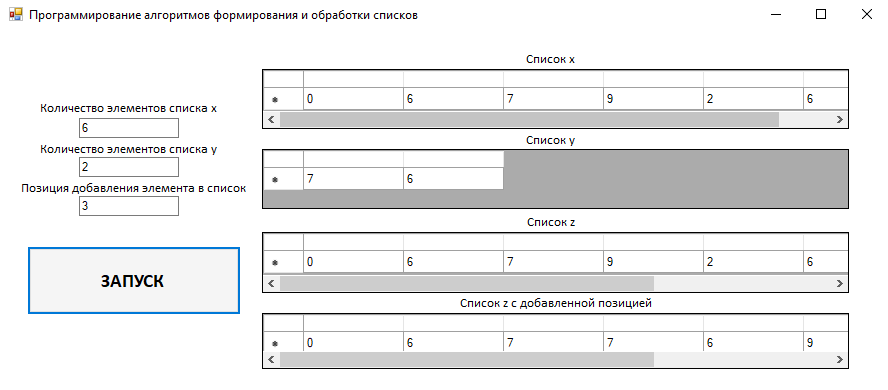
}

}

}

}

**Результат работы программы:**



**Рис.1.** Результат работы Windows Form