**Отчет ПО ЛАБОРАТОРНЫМ РАБОТАМ**

**По информатике**

Лабораторная работа №4.

Обработка одномерных массивов

(наименование лабораторной работы в соответствии с учебным планом)

***Цель и задачи работы:***

Изучение алгоритмов формирования и обработки одномерных

массивов, программирование и отладка программ формирования и обработки

массивов.

***Задание №1***

Обработка одномерных массивов.

В одномерном массиве, состоящем из n целочисленных элементов, вычислить:

1. Номер максимального элемента массива.

2. Произведение элементов массива, расположенных между первым и вторым

нулевыми элементами.

Преобразовать массив таким образом, чтобы в первой его половине располагались

элементы, стоявшие в нечетных позициях, а во второй половине - элементы, стоявшие

в четных позициях.

***Алгоритм решения:***

Блок-схема алгоритма решения задачи представлена ниже.

**Блок-схемы отдельных функций:**

1) max (int A[], int size):

Начало

Max = A[0]

i = 0

нет

i < size

да

Return max

нет

Max < A[i]

i++

да

Конец

Max = A[i]

2) sort (int A[], int size):

Начало

l++

i +=2

B[l] = A[i]

i = 0; i < size

l = 0

B = new int[size]

нет

да

i = 1; i < size

i = 0; i < size

нет

нет

да

Конец

i++

Вывод B[i]

l++

i +=2

B[l] = A[i]

да

3) proiz (int A[], int size):

Начало

да

да

да

да

нет

нет

нет

нет

i++

break

composition \*= A[i]

l++

l == 2

A[i] != 0

&&

l == 1

A[i] == 0

Return composition

Конец

i < size

l = 0

composition = 1

***Листинг программы:***

1) C++

(main)

#include <iostream>

#include "Task1.h"

using namespace std;

int main() {

int size;

cout << "Enter the number of elements in the array: ";

cin >> size;

cout << "\nFill in the array: ";

int\* A = new int[size];

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

cout << "\nA[" << i << "]" << " = ";

cin >> A[i];

}

metka:

cout << "\n1.Finding the maximum. \n2.Sort by position. \n3.Composition between zero elements. \n4.Exit. \n";

int choose;

cin >> choose;

switch (choose) {

case 1:

cout << "\nmax = " << max(A, size);

goto metka;

case 2:

proiz(A, size);

goto metka;

case 3:

cout << "\nComposition = " << proiz(A, size);;

goto metka;

case 4:

break;

}

}

(header)

#pragma once

int max(int A[], int size);

void sort(int A[], int size);

int proiz(int A[], int size);

(Source)

#include "Task1.h"

#include <iostream>

using namespace std;

int max(int A[], int size) {

int max = A[0];

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

if (max < A[i]) max = A[i];

}

return max;

}

void sort(int A[], int size) { // 0 1 2 3 4 5 6 7 8 9

int\* B = new int[size];

int l = 0;

for (int i = 0; i < size; i += 2) {

B[l] = A[i];

l++;

}

for (int i = 1; i < size; i += 2) {

B[l] = A[i];

l++;

}

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

A[i] = B[i];

cout << "\n" << A[i];

}

}

int proiz(int A[], int size) {

int l = 0;

int composition = 1;

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

if (A[i] == 0) l++;

if ((A[i] != 0) && (l == 1)) composition = composition \* A[i];

if (l == 2) break;

}

return composition;

}

2) Python

1) A = [0] \* 5

A[0] = 0

A[1] = 20

A[2] = 40

A[3] = 15

A[4] = 35

max = A[0]

for i in range(len(A)):

if max < A[i]:

max = A[i]

print (max)

2) A= [0] \* 5

A[0] = 0

A[1] = 5

A[2] = 3

A[3] = 0

A[4] = 10

l = 0

composition = 1

for i in range(len(A)):

if A[i] == 0:

l+=1

if (A[i] != 0 and l == 1):

composition = composition \* A[i]

if l == 2:

break;

print (composition)

3) A = [0] \* 5

B = [0] \* 5

A[0] = 1

A[1] = 2

A[2] = 3

A[3] = 4

A[4] = 5

l = 0

i = 0

while i < len(A):

B[l] = A[i]

l = l + 1

i = i + 2

i = 1

while i < len(A):

B[l] = A[i]

l = l + 1

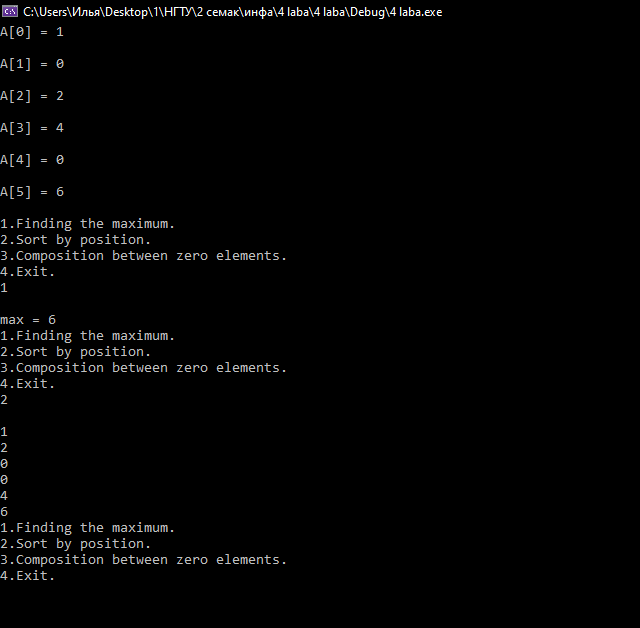
i= i + 2

i = 0

for i in range(len(A)):

print (B[i])

***Пример работы программы:***



*(рис. 1 – пример работы программы на C++)*

**Задание №2.**

Модификация одномерных массивов.

Будем считать, что введённый массив – это рост студентов в группе. Представим, что

все эти студенты пришли на урок физкультуры и встали в шеренгу по росту (человек с

меньшим ростом имеет больший номер). Однако один студент опоздал, и теперь ему

необходимо определить под каким номером он будет стоять в шеренге. После ввода

всех n элементов массива вводится ещё одно число x – рост опоздавшего студента.

Если есть студенты одного роста с опоздавшим, то он становится в шеренгу после них.

Вывести на экран найденный номер (нумерация начинается с 1).

***Алгоритм решения:***

Блок-схема алгоритма решения задачи представлена ниже.

Начало

да

да

нет

нет

i++

number++

Height <= A[i]

Конец

Вывод number

Ввод количества студентов (n) и их рост (A[i]).

Ввод роста опоздавшего (height)

number = 1

i = 1; i <= n

***Листинг программы:***

1) C++

#include <iostream>

using namespace std;

int main() {

cout << "Enter number of students: ";

int n;

cin >> n;

int\* A = new int[n];

cout << "\nEnter the height of each student: ";

for (int i = 1; i <= n; i++) {

cout << "\nStudent N" << i << " : ";

cin >> A[i];

}

cout << "\nEnter the height of a late student: ";

int height;

cin >> height;

int number = 1;

for (int i = 1; i <= n; i++) {

if (height <= A[i]) number++;

}

cout << "\nThe latecomer's number will be: " << number;

}

2) Python

A=[0]\*5

A[0] = 172

A[1] = 188

A[2] = 163

A[3] = 192

A[4] = 178

print("Enter the height of a late student:")

height = int(input())

i = 1

number = 1

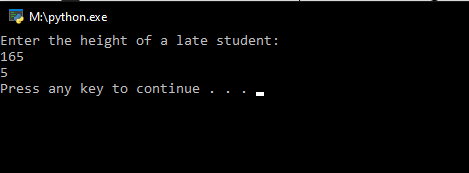
for i in range(len(A)):

if(height <= A[i]):

number = number + 1

print(number)

***Пример работы программы:***



*(рис. 2 – пример работы программы на Python)*

**Задание №3:**

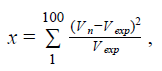
Задание выполняется на языке C++. В соответствии с вариантом необходимо

описать распределение генератора, с помощью этого генератора заполнить 3 массива

размером 50, 100 и 1000 соответственно числами от 1 до 100.

Рассчитать значение критерия хи-квадрат для каждой из выборок в соответствии

с формулой



где *V n* - количество полученных значений n,

*n* - значения внутри выборки ( 1 ≤ *n* ≤ 100 ),

*V exp* - ожидаемой количество значений *n* в рамках выборки

Проанализировать значение критерия хи-квадрат.

Распечатать математическое ожидание каждой выборки и идеальное

математическое ожидание распределения с указанными параметрами. Распечатать

значение критерия хи-квадрат для каждой выборки.

***Листинг программы:***

1)C++

#include <iostream>

#include <random>

#include <vector>

#include <ctime>

#include <cmath>

using namespace std;

const double eps = 0.1e-5;

void Get\_Arr(int msg[], int size, int max\_select)

{

srand(time(NULL));

minstd\_rand0 generator(rand());

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

{

msg[i] = generator() % max\_select + 1;

}

}

double func(double x)

{

return exp(-x \* x / 2);

}

double simpsonIntegr(double a, double b)

{

return ((b - a) / 6) \* (func(a) + 4 \* func((a + b) / 2) + func(b));

}

double LaplasFunction(double x, double eps = ::eps)

{

double a = 0.0;

double b = x;

double tst, tst1, incr;

unsigned int c = 2;

tst1 = simpsonIntegr(a, b);

do {

tst = tst1;

tst1 = 0;

incr = (b - a) / c;

for (unsigned int i = 0; i < c; ++i)

{

tst1 = tst1 + simpsonIntegr(a + incr \* i, a + incr \* (i + 1));

}

c += 1;

} while (fabs(tst - tst1) >= eps);

return tst1;

}

int cou(int msg[], int size, int n, int k)

{

int i = 0;

int cnt = 0;

for (i; i < k; i++)

{

if (msg[i] <= k && msg[i] >= n) cnt++;

}

k += 4;

n += 4;

return cnt;

}

double hi(int msg[], int size)

{

double M = 51, A = 0, sigma = M / 3;

cout << "viborka iz " << size << " elementov" << endl;

int Vn = 0;

double Vexp, hiqs = 0;

int n = 1, k = 4;

for (int i = 1; i < 25; i++)

{

Vn = cou(msg, size, n, k);

Vexp = (LaplasFunction((k - M) / sigma) - LaplasFunction((n - M) / sigma)) \* size;

hiqs += pow((Vn - Vexp), 2) / Vexp;

n += 4;

k += 4;

Vn = 0;

}

return hiqs;

}

double rlmo(int msg[], int size)

{

int i = 1;

double e = 0;

double x = 0;

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

{

x += (double)msg[i];

}

e = x / size;

return e;

}

int main()

{

double f = 36.41503;

int n = 100;

int\* Arr;

Arr = new int[50];

Get\_Arr(Arr, 50, n);

cout << "pri raspredelenii 0.05 ";

if (hi(Arr, 50) > f) cout << "ne norm" << endl;

else cout << "norm";

cout << "ozhidaemoe = " << " 51 " << endl;

cout << "received = " << rlmo(Arr, 50) << endl << endl;

Arr = new int[100];

Get\_Arr(Arr, 100, n);

cout << "pri raspredelenii 0.05 ";

if (hi(Arr, 100) > f) cout << "ne norm" << endl;

else cout << "norm";

cout << "ozhidaemoe = " << " 51 " << endl;

cout << "received = " << rlmo(Arr, 100) << endl << endl;

Arr = new int[1000];

Get\_Arr(Arr, 1000, n);

cout << "pri raspredelenii 0.05 ";

if (hi(Arr, 1000) > f) cout << "ne norm" << endl;

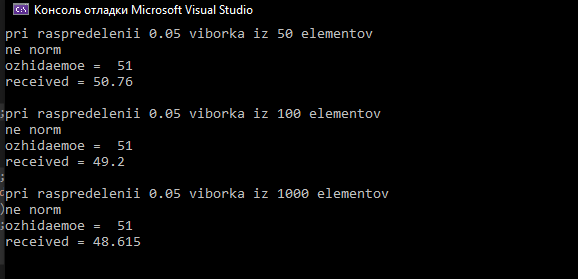
else cout << "norm";

cout << "ozhidaemoe = " << " 51 "<< endl;

cout << "received = " << rlmo(Arr, 1000) << endl << endl;

}

***Пример работы программы:***



(*рис. 3 – пример работы программы на C++*)

**Задание №4.**

Выполнить программную реализацию метода «шаг младенца, шаг великана» и

алгоритма исчисления порядка и решить с помощью компьютера следующие

уравнения:



***Алгоритм решения:***

Блок-схема алгоритма решения задачи представлена ниже.

mod (number, degree, p):

Начало

number

degree

prime number

result = 1

i = 1

degree = degree % (prime -1)

number = number % p

i <= degree

нет

Вывод result

да

result = result \* a

result = result %p

Конец

i = i +1

Начало

да

да

да

да

да

да

нет

нет

нет

нет

нет

Конец

Вывод j \* m - l

i++

break

v++

j = i

l = v

break

j != 0

A[i] == B[v]

j != 0

v = 1;v < 300

i = 1;i < 300

B[i] = mod(a, i, p)

B[i] \*= mod(b, 1, p)

B[i] = mod(B[i], 1, p)

i++

i = 1;i < 300

i = 1;i < 300

нет

да

A[i] = mod(a,i\*m,p)

i++

a = 2

p =30803

b = 28620

k = sqrt(p) + 1

A[300]

B[300]

j = 0

l = 0

***Листинг программы:***

1)C++

#include <iostream>

#include <cmath>

using namespace std;

int mod(int number, int degree, int prime) {

degree = degree % (prime - 1);

number = number % prime;

int result = 1;

for (int i = 1; i <= degree; i++) {

result = result \* number;

result = result % prime;

}

return result;

}

int main() {

cout << "Value finding program: 2^x mod 30803 = 16190" << "\na = 2; p = 30803; b = 16190";

int a = 2; // 3

int p = 30539; //30803 // 61 //11

int b = 28620; // 16190 // 45 // 5

int k = sqrt(p) + 1;

int m = k;

cout << "\nFind the values of m and k" << endl;

cout << "m = k = " << m << endl;

cout << "Find the values of the series: (a^(i\*m)) mod p";

int A[300];

for (int i = 1; i < 300; i++) {

A[i] = mod(a, i \* m, p);

}

int B[300];

int j = 0;

int l = 0;

cout << "\nFind the values of the series: (a^(m) \* b) mod p";

for (int i = 1; i < 300; i++) {

B[i] = mod(a, i, p);

B[i] \*= mod(b, 1, p);

B[i] = mod(B[i], 1, p);

}

for (int i = 1; i < 300; i++) {

for (int v = 1; v < 300; v++) {

if (A[i] == B[v]) {

j = i;

l = v;

}

if (j != 0) break;

}

if (j != 0) break;

}

cout << "\nj = " << j << " l = " << l << endl;

cout << "Find x: x = j \* m - l = " << j \* m - l;

}

2) Python

a = 2

p = 30539

b = 28620

k = pow(p,0.5) + 1

A = [0]\*300

for i in range(len(A)):

A[i] = pow(a,i\*k) % p

B = [0]\*300

j = 0

l = 0

for i in range(len(B)):

B[i] = pow(a,i) % p

B[i] \*= pow(b,1) % p

B[i] = pow(B[i],1) % p

i = 1

v = 1

for i in range(len(A)):

for v in range(len(B)):

if (A[i] == B[v]):

j = i

l = v

if (j != 0):

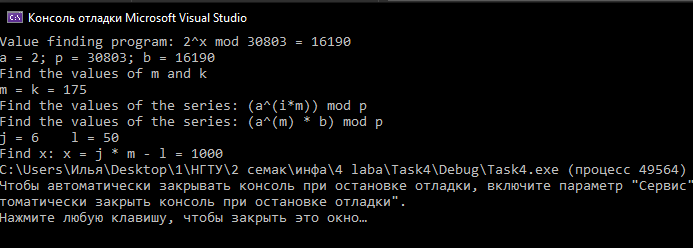
break

if (j != 0):

break

print("x = j \* m - l = ", j \* m - l)

***Пример работы программы:***



(*рис. 4 – пример работы программы на С++*)