МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ государственное БЮДЖЕТНОЕ

образовательное учреждение

высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХ™НИЧЕСКИЙ УНИВЕРСИТЕТ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра защиты информации



**ЛАБОРАТОРНАЯ РАБОТА №4**

«Последовательные одномерные контейнеры»

по дисциплине: «*Программирование*»

|  |  |
| --- | --- |
| Выполнил:  Студент гр. «АБс-323», «АВТФ»  *Раков Михаил Андреевич*  «27» мая 2024г  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | Проверил:  *Ассистент кафедры ЗИ*  *Исаев Глеб Андреевич*  «08» июня 2024г  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

Новосибирск 2024

**Цели и задачи работы:** изучение алгоритмов формирования и обработки одномерных массивов и последовательных контейнеров, программирование и отладка программ формирования и обработки массивов.

**Задание к работе:** написать программу решения задачи в соответствии с индивидуальным вариантом.

**Индивидуальное задание (вариант 22)**:

**Задание 1.**

Отделите корни заданного уравнения, согласно варианту из табл.1, и уточните их одним из методов с точностью до ε = 10^(-4). Решить уравнения методом половинного деления, методом Ньютона и методом простых итераций. Либо другими методами, перечисленными в теоретической части к заданию 1.

*Уравнение: x2 + ex = 2.*

**Задание 2.**

Для генерации чисел использовать алгоритм: minstd\_rand0.

1. Напишите программу, в которой определен массив arr1 из n чисел (15 ≤ n ≤ 30) и инициализирован целыми случайными числами из диапазона [10, 30]. Определите массив arr2 из 5 чисел и инициализируйте массив arr2 целыми случайными числами из диапазона [10, 30].
2. Выведите количество чётных элементов массива 𝑎𝑟𝑟1 на нечётных местах, а также сами элементы.
3. Определите, какие числа из массива arr2 встречаются в массиве arr1 Выведите эти числа и сколько раз они встречаются в массиве arr1 Если ни одно число не встречается, то выведите соответствующее сообщение.

4. Напишите программу, модифицирующую массив случайных символов размера n, как в примере:

[v,!,N,R,(,3] → [!,N,R,(,3,v].

5. Определите и инициализируйте массив, состоящий из случайных целых чисел, входящих в диапазон [1000, 9000]. Вычислите сумму элементов массива. Напишите программу, меняющую местами цифры в элементах массива в порядке возрастания (например, 8315 → 1358). Отсортируйте массив в порядке возрастания. Вычислите сумму элементов массива и сравните с суммой элементов исходного массива.

**Задание 3.**

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.

Использовать критерий хи-квадрат для проверки гипотезы о нормальном распределении выборки. Значение критерия хи-квадрат для каждой из выборок рассчитывается по формуле:



где Vn – количество элементов, попавших в i-ый интервал; Vexp – ожидаемое число попаданий в i-ый интервал.

Проанализировать значение критерия хи-квадрат и вывести:

1) результат проверки гипотезы;

2) ожидаемое и реальное математическое ожидание.

Для генерации чисел использовать алгоритм: ***minstd\_rand0***.

**Задание 4.**«Дилемма заключенного».

На протяжении игровой сессии сражаются 2 алгоритма. В каждом раунде каждый алгоритм выбирает, либо сотрудничество, либо предательство. Если алгоритм А выбирает предательство и алгоритм Б выбирает предательство они получают по 4 очка. Если алгоритм А выбирает сотрудничество, а алгоритм Б выбирает предательство - алгоритм А получает 0 очков, а алгоритм Б получает 20 очков. Если оба алгоритма выбирают сотрудничество оба получают 24 очка. Каждому алгоритму в каждом раунде известны результаты всех предыдущих раундов текущей игровой сессии, на основе этих данных алгоритм может выбирать будет он сотрудничать или предаст.

**Задание 5.**

Задание выполняется на языке C++. Для генерации чисел использовать алгоритмы: енератор псевдослучайных чисел на основе алгоритма BBS**.**

1. **Листинг программы**

**Язык C++**

#include <iostream>

#include <fstream>

#include <iomanip>

#include <vector>

#include <cmath>

#include <tuple>

using namespace std;

using Output\_data = tuple<double, double, double>;

using Outs = vector<Output\_data>;

double f (double x)

{

return pow(x, 2) + pow(M\_E, x) - 2;

}

double deriv\_f (double x)

{

return 2 \* x + pow(M\_E, x);

}

pair<double, Outs>

method\_bisection (double a, double b, double eps, size\_t maxIters)

{

Outs outs;

double result = 0;

for (size\_t i = 0; i < maxIters and b-a >= eps; ++i)

{

double c = (a + b) / 2;

result = c;

outs.emplace\_back(a, b, result);

if (f(c) == 0)

break;

(f(a) \* f(c) < 0 ? b : a) = c;

}

return {result, outs};

}

pair<double, Outs>

method\_Newton (double x0, double eps, size\_t maxIters)

{

Outs outs;

double result = 0;

for (size\_t i = 0; i < maxIters; ++i)

{

double xNext = x0 - f(x0) / deriv\_f(x0);

result = xNext;

outs.emplace\_back(xNext, x0, result);

if (fabs(xNext - x0) < eps)

break;

x0 = xNext;

}

return {result, outs};

}

pair<double, Outs>

method\_8 (double x, double xPrev, double eps, size\_t maxIters)

{

Outs outs;

double result = 0;

for (size\_t i = 0; i < maxIters; ++i)

{

double xNext = x - (x - xPrev) \* f(x) / (f(x) - f(xPrev));

result = xNext;

outs.emplace\_back(x, xPrev, result);

if (fabs(xNext - x) < eps)

break;

xPrev = x;

x = xNext;

}

return {result, outs};

}

ostream & operator<< (ostream& out, Outs const& v)

{

for (auto [a, b, r] : v)

{

out << setw(10) << left << a

<< setw(10) << left << b

<< setw(10) << left << r

<< endl;

}

return out;

}

int main()

{

system("chcp 65001");

double a = -2, b = -0.01, c = 1;

ofstream out("file.txt");

Outs outs, outs2;

double res, res2;

out << fixed << setprecision(4);

tie(res, outs) = method\_bisection (a, b, 0.001, 100);

tie(res2, outs2) = method\_bisection (b, c, 0.001, 100);

out << "Решение методом половинного деления.\n"

<< "Корни: " << res << " " << res2 << endl

<< outs << endl << outs2 << endl;

tie(res, outs) = method\_Newton (a, 0.001, 100);

tie(res2, outs2) = method\_Newton (b, 0.001, 100);

out << "Решение методом Ньютона.\n"

<< "Корни: " << res << " " << res2 << endl

<< outs << endl << outs2 << endl;

tie(res, outs) = method\_8 (a, b, 0.001, 100);

tie(res2, outs2) = method\_8 (b, c, 0.001, 100);

out << "Решение комбинированным методом секущих-хорд.\n"

<< "Корни: " << res << " " << res2 << endl

<< outs << endl << outs2 << endl;

out.close();

}

**Язык Haskell**

import Data.Tuple (Tuple (..))

import Text.Printf

newtype Output\_data = Output\_data Double Double Double deriving (Eq, Show)

type Outs = [Output\_data]

bisection :: Double -> Double -> Double -> Int -> Outs

bisection a b eps maxIters = go [] a b

where

go os l r

| r - l < eps || maxIters <= 0 = map (uncurry Output\_data) os

| otherwise =

let c = (l + r) / 2

in case signum (f l) == signum (f c) of

False -> go ((c, r, c) : os) l c

True -> go ((l, c, c) : os) c r

newton :: Double -> Double -> Int -> Outs

newton x eps maxIters = go x []

where

go xn os

| abs (xn - x) < eps || maxIters <= 0 = map (uncurry Output\_data) (reverse os)

| otherwise = go xNext ((x, xn, xNext) : os)

where

xNext = xn - f xn / deriv\_f xn

combined :: Double -> Double -> Double -> Int -> Outs

combined a b eps maxIters = go [] a b

where

go os x0 x1

| x1 - x0 < eps || maxIters <= 0 = map (uncurry Output\_data) os

| otherwise =

let xNext = x0 - (x0 - x1) \* f x0 / (f x0 - f x1)

in go ((x0, x1, xNext) : os) x1 xNext

f, deriv\_f :: Double -> Double

f = (^ 2) + exp

deriv\_f x = 2 \* x + exp x

signum :: Double -> Double

signum x = if x > 0 then 1.0 else if x < 0 then -1.0 else 0.0

main :: IO ()

main = do

let a = -2

let b = -0.01

let c = 1 :: Double

let eps = 0.001

let maxIters = 100 :: Int

let outs = bisection a b eps maxIters

let outs2 = bisection b c eps maxIters

putStrLn $ "Решение методом половинного деления."

putStrLn $ "Корни:" ++ show (fst outs) ++ " " ++ show (fst outs2)

mapM\_ print outs

putStrLn ""

mapM\_ print outs2

putStrLn ""

let outs = newton a eps maxIters

let outs2 = newton b eps maxIters

putStrLn $ "Решение методом Ньютона."

putStrLn $ "Корни:" ++ show (fst outs) ++ " " ++ show (fst outs2)

mapM\_ print outs

putStrLn ""

mapM\_ print outs2

putStrLn ""

let outs = combined a b eps maxIters

let outs2 = combined b c eps maxIters

putStrLn $ "Решение комбинированным методом секущих-хорд."

putStrLn $ "Корни:" ++ show (fst outs) ++ " " ++ show (fst outs2)

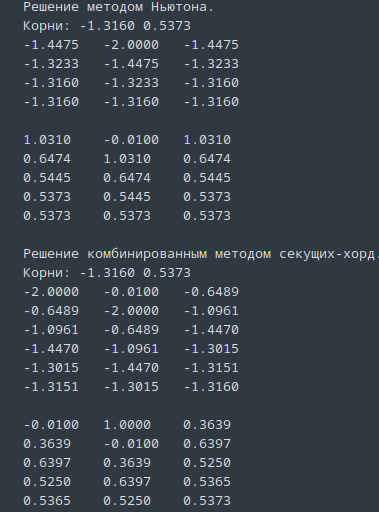
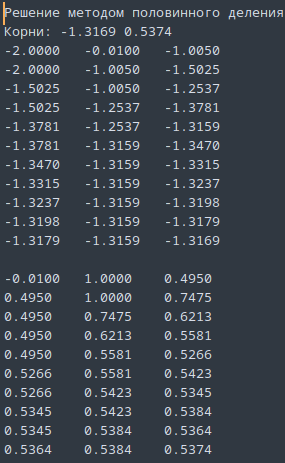
mapM\_ print outs

putStrLn ""

mapM\_ print outs2

putStrLn ""

**Результат работы программы**

****

1. **Листинг программы**

**Язык C++**

#include <iostream>

#include <fstream>

#include <vector>

#include <cmath>

#include <random>

#include <algorithm>

using namespace std;

minstd\_rand0 gen;

vector<int> random\_array (size\_t length, int minV, int maxV)

{

vector<int> arr(length);

for (auto& el : arr)

{

el = minV + (gen() % (maxV - minV));

}

return arr;

}

template <class T>

ostream & operator<< (ostream& out, vector<T> const& v)

{

for (auto el : v)

{

out << el << " ";

}

return out;

}

vector<int> task2 (vector<int> const& v)

{

vector<int> res;

for (size\_t i = 0; i < v.size(); i += 2)

{

if (v[i] % 2 == 0)

res.push\_back(v[i]);

}

return res;

}

void task3 (vector<int> const& arr1, vector<int> const& arr2)

{

vector<int> res;

copy\_if (arr2.begin(), arr2.end(), back\_inserter(res), [&arr1] (int i) {

return count(arr1.begin(), arr1.end(), i) != 0;

});

if (res.empty())

{

cout << "Ни одно число из arr2 в arr1 не встречается." << endl;

return;

}

cout << "В arr1 встречаются следующие элементы из arr2: " << endl;

for (auto el : res)

{

cout << el << " ("

<< count(arr1.begin(), arr1.end(), el)

<< ") ";

}

cout << endl;

}

vector<char> task4 (vector<char> v)

{

rotate(v.begin(), v.begin() + 1, v.end());

return v;

}

int f (int i)

{

int digits[4];

for (auto& d :digits)

{

d = i % 10;

i /= 10;

}

auto [d, c, b, a] = digits;

return c \* 1000

+ b \* 100

+ d \* 10

+ a;

}

void task5(vector<int> const& v)

{

vector<int> v2 = v;

for (auto& el : v2)

{

el = f(el);

}

cout << endl << v << endl << v2 << endl;

sort(v2.begin(), v2.end());

int sumV = accumulate(v.begin(), v.end(), 0);

int sumV2 = accumulate(v2.begin(), v2.end(), 0);

cout << "Сумма элементов v (" << sumV << ") "

<< (sumV > sumV2 ? ">" : "<")

<< " суммы элементов v2 (" << sumV2 << ")."

<< endl;

}

int main()

{

system("chcp 65001");

size\_t n;

cin >> n;

vector<int> arr1 = random\_array(n, 15, 31);

vector<int> arr2 = random\_array(5, 15, 31);

cout << arr1 << endl << arr2 << endl << endl;

vector<int> res2 = task2(arr1);

cout << "Чётные элементы массива arr1 на нечётных местах: "

<< res2 << endl <<"Их количество: "

<< res2.size() << endl << endl;

task3(arr1, arr2);

vector<char> v4 = {'v', '!', 'N', 'R', '(', '3'};

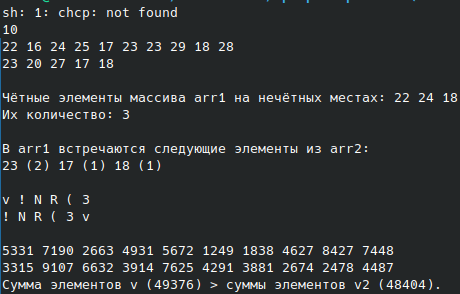
cout << endl << v4 << endl

<< task4(v4) << endl;

task5(random\_array(10, 1000, 9001));

}

**Результат работы программы**



1. **Листинг программы**

**Язык C++**

#include <iostream>

#include <numeric>

#include <random>

#include <cmath>

#include <iomanip>

using namespace std;

minstd\_rand0 gen;

vector<int> random\_array (size\_t length, int minV, int maxV)

{

vector<int> arr(length);

for (auto& el : arr)

{

el = minV + (gen() % (maxV - minV));

}

return arr;

}

double real\_math\_expectation (vector<int> const& v)

{

return accumulate (v.begin(), v.end(), 0.0) / v.size();

}

double p\_table (int df)

{

static double his [] = {

3.84, 5.99, 7.81, 9.49, 11.07,

12.59, 14.07, 15.51, 16.92, 18.31

};

return his[df-1];

}

double hi\_square (vector<int> const& v, int minV, int maxV, size\_t numIntv)

{

double expCnt = v.size() / static\_cast<double>(numIntv);

vector<int> intervals (numIntv, 0);

for (auto el : v)

{

size\_t ind = numIntv \* (el - minV) / (maxV - minV + 1);

++intervals[ind];

}

return accumulate (intervals.begin(), intervals.end(), 0.0,

[expCnt] (double prev, int nCnt)

{

return prev + pow(nCnt - expCnt, 2) / expCnt;

}

);

}

int main()

{

system("chcp 65001");

size\_t df = 9;

vector<vector<int>> arrays =

{

random\_array(50, 1, 100),

random\_array(100, 1, 100),

random\_array(1000, 1, 100)

};

for (auto const& arr : arrays)

{

double hi = hi\_square(arr, 1, 100, df);

cout << "Ожидаемое математическое ожидание: " << df << endl

<< "Реальное математическое ожидание: "

<< real\_math\_expectation(arr) << endl

<< "Значение хи-квадрат: " << hi << endl

<< "Гипотеза о нормально распределении "

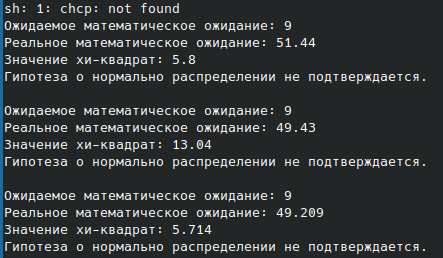
<< (hi < p\_table(df) ? "не" : "") << " подтверждается." << endl;

cout << endl;

}

}

**Результат работы программы**



1. **Листинг программы**

**Язык C++**

#include <iostream>

#include <random>

#include <vector>

#include <cstdint>

#include <algorithm>

using namespace std;

using Alg\_type = bool (\*) (int32\_t, vector<bool>, vector<bool>);

bool play\_game(Alg\_type a1, Alg\_type a2, int num\_rounds)

{

int score1 = 0;

int score2 = 0;

vector<bool> self\_choices1;

vector<bool> self\_choices2;

vector<bool> enemy\_choices1;

vector<bool> enemy\_choices2;

for (int i = 0; i < num\_rounds; ++i)

{

bool choice1 = a1(i, self\_choices1, enemy\_choices1);

bool choice2 = a2(i, self\_choices2, enemy\_choices2);

self\_choices1.push\_back(choice1);

self\_choices2.push\_back(choice2);

enemy\_choices1.push\_back(choice2);

enemy\_choices2.push\_back(choice1);

if (choice1 == choice2)

{

score1 += (choice1 ? 24 : 4);

score2 += (choice1 ? 24 : 4);

}

else

{

(choice1 ? score2 : score1) += 20;

}

}

return score1 > score2;

}

bool alg1(int32\_t round\_number, vector<bool> self\_choices, vector<bool> enemy\_choices)

{

if (round\_number == 0)

return true;

return enemy\_choices[round\_number - 1];

}

bool alg2(int32\_t round\_number, vector<bool> self\_choices, vector<bool> enemy\_choices)

{

return false;

}

bool alg3(int32\_t round\_number, vector<bool> self\_choices, vector<bool> enemy\_choices)

{

bool anyBetrayed = any\_of(enemy\_choices.begin(), enemy\_choices.end(), [] (bool b) {

return b;

});

return not anyBetrayed;

}

int main() {

// Запустить игру

bool res1 = play\_game(alg1, alg2, 100);

bool res2 = play\_game(alg1, alg3, 100);

bool res3 = play\_game(alg2, alg3, 100);

cout << "Первый бой выиграл " << (res1 ? "alg1" : "alg2") << endl

<< "Второй бой выиграл " << (res2 ? "alg1" : "alg3") << endl

<< "Третий бой выиграл " << (res3 ? "alg2" : "alg3") << endl;

}

**Результат работы программы**



1. **Листинг программы**

**Язык C++**

#include <iostream>

#include <tuple>

#include <cmath>

#include <vector>

using namespace std;

long long pow\_mod(long long a, long long x, long long p)

{

long long result = 1;

while (x > 0)

{

if (x % 2 == 1)

result = result \* a % p;

a = a \* a % p;

x /= 2;

}

return result;

}

pair <long long, long long>

pseudo\_rand\_BBS (long long x\_n, long long M)

{

long long rndS = 0;

for (int i = 0; i < sizeof(long long) \* 8; ++i)

{

x\_n = pow\_mod (x\_n, 2, M);

rndS = rndS \* 2 + (x\_n % 2);

}

return {rndS, x\_n};

}

pair <vector<long long>, long long>

pseudo\_rand\_array (size\_t len, long x\_n, long long M)

{

vector<long long> res (len);

long long r, x = x\_n;

for (auto& el : res)

{

tie(r, x) = pseudo\_rand\_BBS(x, M);

el = r;

}

return {res, x};

}

int main()

{

long long p = 6025631;

long long q = 8478859;

auto [rndVec, gen] = pseudo\_rand\_array(10, 1234, p\*q);

for (auto el : rndVec)

{

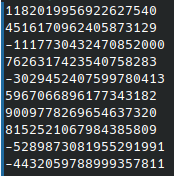
cout << el << endl;

}

cout << endl;

}

**Результат работы программы**



**Вывод**

В ходе данной лабораторной работы я изучил алгоритмы формирования и обработки одномерных массивов и последовательных контейнеров, программирование и отладку программ формирования и обработки массивов.