Лабораторная работа №3

“Экспериментальное определение количества элементарных операций языка высокого уровня в программной реализации алгоритма”

1 ЦЕЛЬ РАБОТЫ

Экспериментальная проверка теоретически полученной функции трудоемкости для алгоритма точного решения задачи о сумме методом полного перебора. Получение практических навыков расстановки счетчика операций в программе на языке высокого уровня.

2 ПОСТАНОВКА ЗАДАЧИ

1. Составить структурную схему алгоритма точного решения задачи о сумме.

2. Включить в функцию TaskSum строки счетчика элементарных операций в соответствии с принятой методикой.

3. Рассмотреть два варианта массива – 1) заполненный вручную; 2) заполненный с помощью генератора случайных чисел.

|  |  |  |  |
| --- | --- | --- | --- |
| Вариант | Размерность вектора случайных чисел | Максимальное случайное число в векторе | Значение суммы (V) |
| 4 | 13 | 90 | 25 |

3 ТЕКСТ ПРОГРАММЫ

#include <iostream>

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <math.h>

using namespace std;

void create\_array(int N, int vector[])

{

srand(time(NULL));

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

vector[i] = rand () % 91;

}

void tasksum(int N, int vector[])

{

int counter = 0;

int V = 25;

int Cnt[N];

int i = 0;

counter += 3;

do

{

Cnt[i] = 0;

i += 1;

}

while (i < N);

counter += 5\*N;

do

{

int Sum = 0;

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

Sum = Sum + vector[i]\*Cnt[N-i];

counter += 1+10\*N;

if (Sum == V)

{

counter++;

for(int i = N; i >= 1; i--)

cout << Cnt[i] << " ";

counter += 1+4\*N;

cout <<endl<< "Число операций: "<<counter<<endl;

return;

}

int j = N;

while (Cnt[j] == 1)

{

Cnt[j] = 0;

j = j-1;

}

Cnt[j] = 1;

counter += 5+6\*N;

}

while (Cnt[0] != 1);

for(int i = N; i >= 1; i--)

cout << Cnt[i] << " ";

cout <<endl<< "Число операций: "<<counter<<endl;

}

int main()

{

int N;

cout << "Размер вектора: "; cin >> N;

cout << "Макс. случайное число в векторе: 90" << endl;

cout << "Значение суммы V: 25" << endl;

int Hn = (8\*N\*pow(2,N)+16\*pow(2,N)-3\*N-12);

cout<<"F^(N): "<<Hn<<endl;

int vector[N];

create\_array(N, vector);

cout<<endl;

cout << "Рандомный вектор:" << endl;

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

cout << vector[i] <<" ";

cout<<endl;

tasksum(N, vector);

system("pause");

cout<<endl;

cout << "Вручную:" << endl;

int vector1[N];

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

cin >> vector1[i];

tasksum(N, vector1);

system("pause");

return 0;

}

4 СТРУКТУРНАЯ СХЕМА

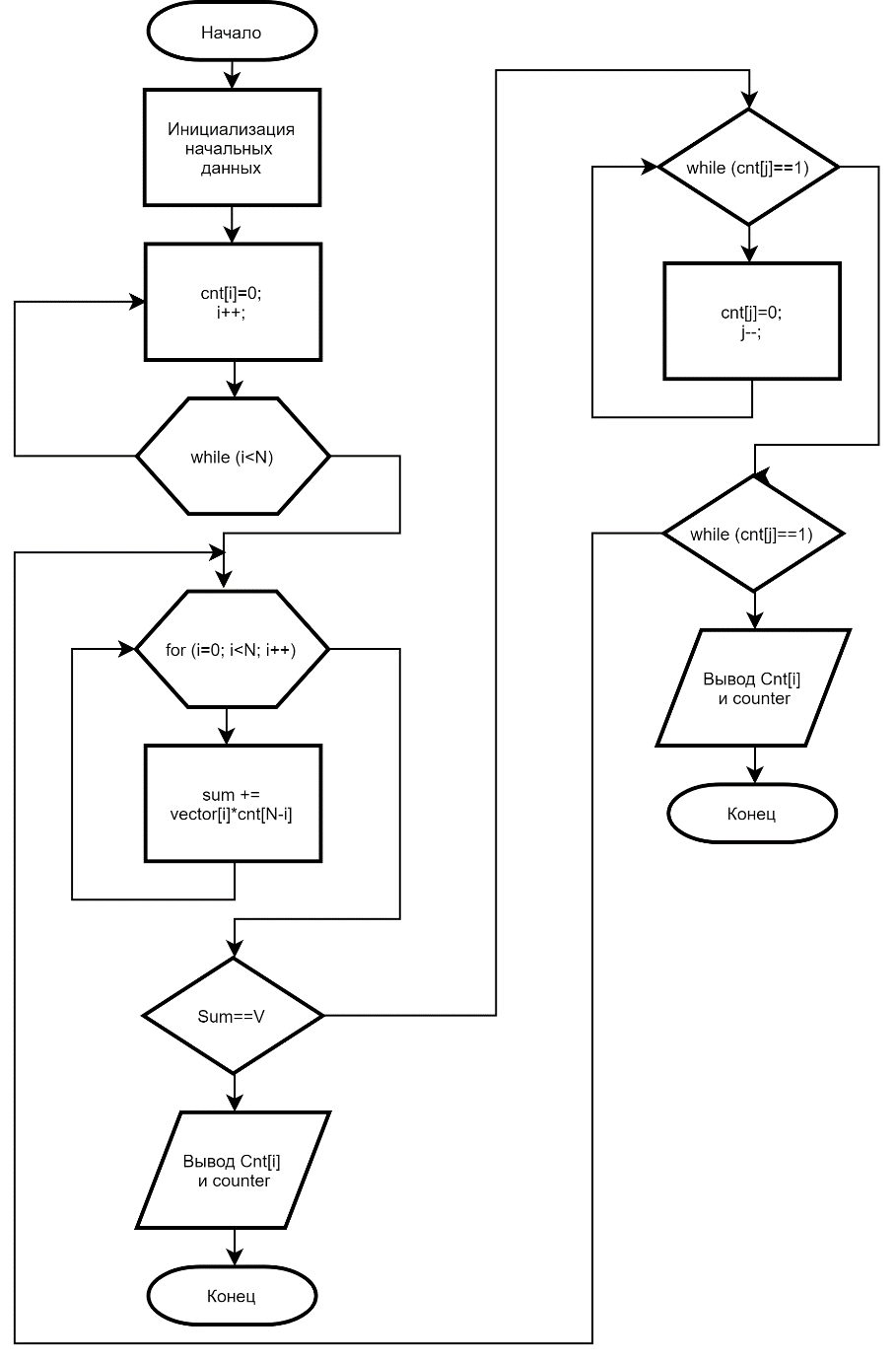


Рисунок 1 – Блок-схема функции Tusksum

5 РЕЗУЛЬТАТЫ

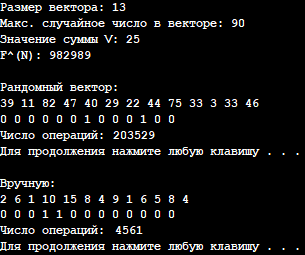


Рисунок 2 – Тест №1

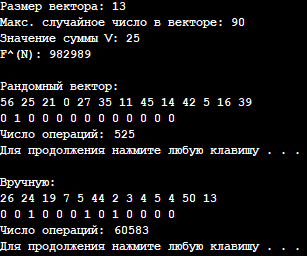


Рисунок 3 – Тест №2

ВЫВОД

В ходе выполнения лабораторной работы была выполнена экспериментальная проверка теоретически полученной функции трудоемкости для алгоритма точного решения задачи о сумме методом полного перебора. Получил практические навыки расстановки счетчика операций в программе на языке высокого уровня.