|  |
| --- |
| **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ** |
| **ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ** |
| **УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ** |
| **«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»** |
|  |
| Институт информационных технологий и управления в технических системах |
| (полное название института) |
|  |
| кафедра «Информационные системы» |
| (полное название кафедры) |

**ОТЧЁТ**

по лабораторной работе №3

на тему

«Экспериментальное определение количества элементарных операций языка

высокого уровня в программной реализации алгоритма»

Вариант 16

Выполнил

студент ИИТУТС

группы ИС/б-18-2-о

Радыгина Екатерина

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | | | | | | | |
|  | (должность, учёная степень преподавателя) | | | | | | | | |
|  |  | | | | | | | | |
|  | (ФИО преподавателя) | | | | | | | | |
|  | « |  | » |  |  |  | 20 | 20 | г. |
|  |  | | | | | | | | |
|  | (оценка) | | | | | | | | |
|  |  | | | | | | | | |
|  |  | | | | | | | | |

Севастополь 2020

**3.1 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.

**3.2 Вариант задания**

Для заданных значений по варианту задания заполнить таблицу 2. Рассмотреть два варианта массива – 1) заполненный вручную; 2) заполненный с помощью генератора случайных чисел.

Таблица 1 – Исходные данные

|  |  |  |  |
| --- | --- | --- | --- |
| Вариант | Размерность вектора случайных чисел | Максимальное случайное число в векторе | Значение суммы (V) |
| 16 | 8 | 100 | 17 |

**3.3 Ход работы**

3.3.1 Составим структурную схему для функции TaskSum (рисунок 3.1)

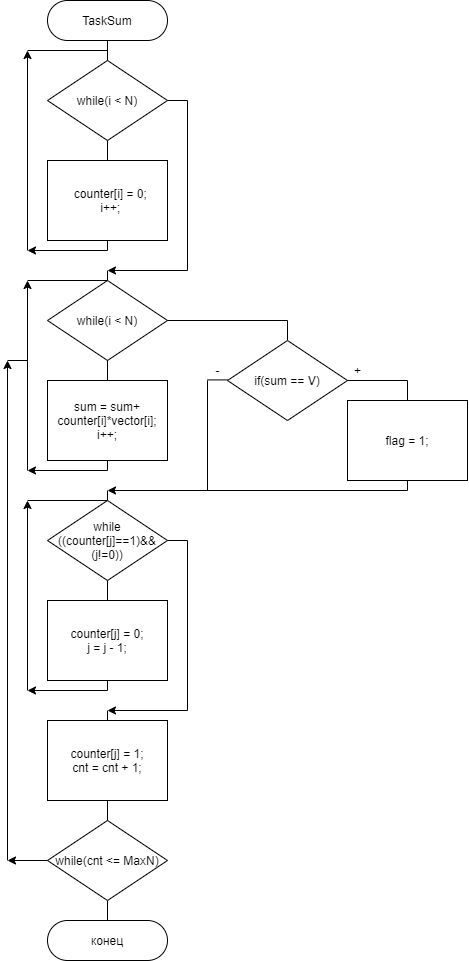


Рисунок 3.1 – Структурная схема функции

3.3.2 Рассмотрим вариант массива, заполненный вручную, и введем полученные данные в таблицу 2.

Таблица 2 – Результат экспериментов ручным заполнением

|  |  |  |  |
| --- | --- | --- | --- |
| N эксперимента | 2n | f(n) = 8\*n\*2n + 16 \*2n – 3\*n -12 | f(n) по данным программного расчёта |
| 1 | 256 | 20444 | 1197 |
| 2 | 256 | 20444 | 673 |
| 3 | 256 | 20444 | 13471 |

На рисунках 3.2 – 3.4 приведены результаты всех экспериментов и входные данные.

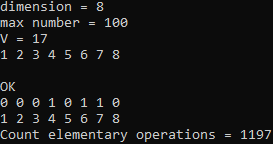


Рисунок 3.2 – Эксперимент 1

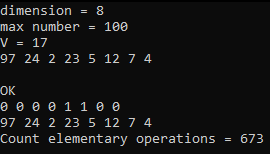


Рисунок 3.3 – Эксперимент 2

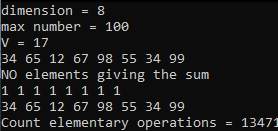


Рисунок 3.4 – Эксперимент 3

3.3.3 Рассмотрим вариант массива, заполненный с помощью генератора случайных чисел, и введем полученные данные в таблицу 3.

Таблица 3 – Результаты экспериментов rand()

|  |  |  |  |
| --- | --- | --- | --- |
| N эксперимента | 2n | f(n) = 8\*n\*2n + 16 \*2n – 3\*n -12 | f(n) по данным программного расчёта |
| 1 | 256 | 20444 | 146 |
| 2 | 256 | 20444 | 13471 |
| 3 | 256 | 20444 | 255 |

На рисунках 3.5 – 3.7 приведены результаты экспериментов с введением чисел через rand().

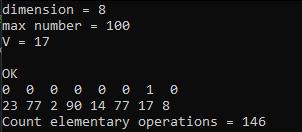


Рисунок 3.5 – Эксперимент 1

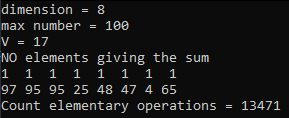


Рисунок 3.6 – Эксперимент 2

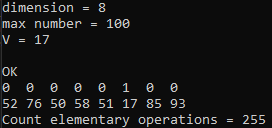


Рисунок 3.7 – Эксперимент 3

Код программы приведен в приложении А.

**Вывод**

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

Исходя из экспериментов, можно сделать вывод, что вариант массива, заполненный с помощью генератора случайных чисел, по большей части выдает набор, для которого задача нерешаема.

Подсчет элементарных операций показал, что практическое количество операций при несуществующем наборе (происходит перебор всех чисел) не совпадает с теоритическим.

**ПРИЛОЖЕНИЕ А**

#include <conio.h>

#include <math.h>

#include <stdio.h>

#include <stdlib.h>

#include <iomanip>

#include <time.h>

#include <iostream>

#define SIZE ;

using namespace std;

int N = 8;

int vector[8];

int counter[8];

int V = 17;

int flag = 0;

int countOp = 0;

void TaskSum(){

int i,j,sum,cnt,MaxN;

flag = 0;

MaxN = int(pow(2,N)-1);

i = 0;

cnt = 1;

countOp += 6;

while(i < N){

counter[i] = 0;

i++;

countOp =+ 3;

}

do{

sum = 0;

i = 0;

j = N - 1;

countOp += 4;

while(i < N){

sum = sum+counter[i]\*vector[i];

i++;

countOp += 5;

}

if(sum == V){

flag = 1;

countOp += 2;

cout << endl;

return;

}

while((counter[j]==1)&&(j!=0)){

counter[j] = 0;

j = j - 1;

countOp += 6;

}

counter[j] = 1;

cnt = cnt + 1;

countOp += 3;

}

while(cnt <= MaxN);countOp++;

}

int main(){

srand(time(0));

int min,cnt,power,sum,j,temp,c;

double result;

cout << "dimension = 8\n"

<< "max number = 100\n"

<< "V = 17\n";

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

fscanf(stdin, "%d", &temp);

//temp = rand() % 100 + 1;

vector[i] = temp;

}

TaskSum();

if(flag == 1) cout << "OK\n";

else cout << "NO elements giving the sum\n";

for(int k = 0; k < N; k++) printf("%d%s",counter[k]," ");

printf("%s\n"," ");

for(int k = 0; k < N; k++) printf("%d%s",vector[k]," ");

cout << endl << "Count elementary operations = " << countOp << endl;

}