Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и информационных технологий

Кафедра вычислительной техники

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №2**

По дисциплине “Алгоритмы и структуры данных”

Вариант 21

|  |  |  |
| --- | --- | --- |
| Преподаватель | \_\_\_\_\_\_\_\_\_\_ | Матковский И.В. |
|  | подпись, дата |  |
| Студент КИ20-07Б, 032052463 | \_\_\_\_\_\_\_\_\_\_ | Базаров А.С. |
|  | подпись, дата |  |
|  |  |  |
|  |  |  |

Красноярск 2021

**СОДЕРЖАНИЕ**

[1 Задание 4](#_Toc81912682)

[2 Исходный код 4](#_Toc81912683)

[3 Результаты 5](#_Toc81912684)

1. Задание

|  |  |  |  |
| --- | --- | --- | --- |
| | **№** | **Задание** | | --- | --- | | **Структуры данных** |
| Разделить числовую последовательность на 2 – из четных и из нечетных элементов исходной. | Очередь/стек |

1. Исходный код

Программа со стеком:

#include <clocale>

#include <iostream>

using namespace std;

struct node // Описание узла

{

int value; // Информационное поле

node\* link; // Поле для связи с другим узлом

};

int main()

{

setlocale(LC\_ALL, "rus");

node\* eventop, \*oddtop, \*evenk, \*oddk;

int w, count; // Буферная переменная

eventop = NULL; // Стек четный пустой

oddtop = NULL; // Стек нечетный пустой

/\* Построение стека \*/

cout << "Сколько элементов вы хотите ввести? ";

cin >> count;

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

cout << "Введите число:";

cin >> w;

if (w % 2 == 0) {

evenk = new node; // Выделение динамической памяти

evenk->link = eventop; // Связь с предыдущим узлом

evenk->value = w; // Записываем в информационное поле введенное число

eventop = evenk; // Вершина стека перемещается на вновь созданный узел

}

else {

oddk = new node; // Выделение динамической памяти

oddk->link = oddtop; // Связь с предыдущим узлом

oddk->value = w; // Записываем в информационное поле введенное число

oddtop = oddk; // Вершина стека перемещается на вновь созданный узел

}

}

cout << "Четный Стек" << endl;

evenk = eventop;

while (evenk != NULL) // Пока не достигли конца стека

{

cout << evenk->value << " ";

evenk = evenk->link; // Перемещение к следующему узлу

}

cout << endl;

cout << "Нечетный Стек" << endl;

oddk = oddtop;

while (oddk != NULL) // Пока не достигли конца стека

{

cout << oddk->value << " ";

oddk = oddk->link; // Перемещение к следующему узлу

}

return 0;

}

Программа с очередью

#include <clocale>

#include <iostream>

using namespace std;

struct node // Описание узла

{

int value; // Информационное поле

node\* link; // Поле для связи с другим узлом

};

int main()

{

setlocale(LC\_ALL, "rus");

node\* evenk; // Рабочий указатель

node\* evenl, \* evenr; // Левый и правый указатели

node\* oddk; // Рабочий указатель

node\* oddl, \* oddr; // Левый и правый указатели

int w; // Буферная переменная

int count, even = 0, odd = 0;

cout << "Сколько элементов вы хотите ввести?\_";

cin >> count;

// Создание пеpвого узла

cout << "Введите четное число:";

cin >> w;

evenk = new node;

evenk->link = NULL;

evenk->value = w;

evenl = evenr = evenk;

cout << "Введите нечетное число:";

cin >> w;

oddk = new node;

oddk->link = NULL;

oddk->value = w;

oddl = oddr = oddk;

// Построение остальных узлов очереди

for (int i = 2; i < count; i++)

{

cout << "Введите число:";

cin >> w;

if (w % 2 == 0) {

evenk = new node;

evenk->link = NULL;

evenk->value = w;

evenr->link = evenk;

evenr = evenk;

}

else {

oddk = new node;

oddk->link = NULL;

oddk->value = w;

oddr->link = oddk;

oddr = oddk;

}

}

// Вывод содержимого очереди

evenk = evenl;

cout << "Четная Очередь" << endl;

while (evenk != NULL)

{

cout << evenk->value << " ";

evenk = evenk->link;

}

evenk = evenl;

cout << endl;

oddk = oddl;

cout << "Нечетная Очередь" << endl;

while (oddk != NULL)

{

cout << oddk->value << " ";

oddk = oddk->link;

}

oddk = oddl;

return 0;

}

1. Теоретические оценки сложности

Теоретическая сложность программы со стеком составляет так как программа разделяет поступающий поток чисел сразу на два стека – четный и нечетный.

Теоретическая сложность программы со очередью составляет так как программа разделяет поступающий поток чисел сразу на две очереди – четная и нечетная.

1. Результаты

Результаты выполнения алгоритма приведены в таблице 1

Таблица 1

|  |  |  |
| --- | --- | --- |
| Количество элементов | Очередь, время в мс. | Стек, время в мс. |
| 10 | 0.0024 | 0.0025 |
| 100 | 0.0204 | 0.0205 |
| 1000 | 0.2004 | 0.2005 |