

Пермский национальный исследовательский политехнический университет

Электротехнический факультет

Информатика и вычислительная техника

ОТЧЕТ

О выполнении лабораторной работы №4 «Работа с одномерными массивами»

Студент: Шадрин И. Д.

Группа: ИВТ-23-1Б

Преподаватель: Яруллин Д. В.

Пермь – 2024

1. Формулировка индивидуального задания

Сформировать одномерный массив целых чисел, используя датчик случайных чисел.

Распечатать полученный массив.

Удалить элементы, индексы которых кратны 3.

Добавить после каждого отрицательного элемента массива элемент со значением $|M[i - 1] + 1|$.

Распечатать полученный массив.

2. Описание использованных типов данных

При выполнении данной лабораторной работы использовался встроенный тип данных `int`, предназначенный для работы с целыми числами.

3. Исходные коды разработанных программ

Листинг 1: Исходные коды программы main.o (файл: main.cpp)

```
#include <iostream>

using namespace std;

void init_array(int *arr, int len);
void print_array(int *arr, int len);
void format_array(int *arr, int *len);

int main() {
    srand(time(nullptr));

    int len = rand() % 20 + 1;

    int *arr = (int *) calloc(len, sizeof(int));

    cout << "Length of array: " << len << endl;
    init_array(arr, len);
    print_array(arr, len);
    format_array(arr, &len);
    print_array(arr, len);
    free(arr);

    return 0;
}

void init_array(int *arr, int len) {
    for (int i = 0; i < len; i++) {
        arr[i] = rand() % 201 - 100;
    }
}

void print_array(int *arr, int len) {
    for (int i = 0; i < len; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
}

void format_array(int *arr, int *len) {
    int step = 1;

    for (int i = 1; i < *len; i++) {
        if (i % 3 == 0) {
            ++step;
        }

        else {
            arr[i - step] = arr[i];
        }
    }
}
```

```
}

*len -= step;
arr = (int *) realloc(arr, *len * sizeof(int));

for (int i = 0; i < *len; i++) {
    if (arr[i] < 0) {
        ++*len;
        arr = (int *) realloc(arr, *len * sizeof(int));

        for (int j = *len - 1; j > i + 1; j--) {
            arr[j] = arr[j - 1];
        }

        arr[i + 1] = abs(arr[i] + 1);
    }
}
}
```

4. Описание тестовых примеров

Таблица 1: Тестовые примеры

Ввод	Ожидаемый вывод	Вывод
10 94 -57 -76 29 18 58 88 41 4 -5	-57 56 -76 75 18 58 41 4	-57 56 -76 75 18 58 41 4
1 84		
9 31 -5 -84 -17 -78 80 -52 52 31	-5 4 -84 83 -78 77 80 52 31	-5 4 -84 83 -78 77 80 52 31

5. Скриншоты

```
root@aw313146638:~/labs/Labs_PSTU/Sem_2/Green/4# g++ -O0 -g3 -o main.o main.cpp
root@aw313146638:~/labs/Labs_PSTU/Sem_2/Green/4# valgrind ./main.o
==30832== Memcheck, a memory error detector
==30832== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==30832== Using Valgrind-3.16.1 and LibVEX; rerun with -h for copyright info
==30832== Command: ./main.o
==30832==
Length of array: 10
94 -57 -76 29 18 58 88 41 4 -5
-57 56 -76 75 18 58 41 4
==30832==
==30832== HEAP SUMMARY:
==30832==   in use at exit: 0 bytes in 0 blocks
==30832==   total heap usage: 2 allocs, 2 frees, 73,728 bytes allocated
==30832==
==30832== All heap blocks were freed -- no leaks are possible
==30832==
==30832== For lists of detected and suppressed errors, rerun with: -s
==30832== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
root@aw313146638:~/labs/Labs_PSTU/Sem_2/Green/4# valgrind ./main.o
==30833== Memcheck, a memory error detector
==30833== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==30833== Using Valgrind-3.16.1 and LibVEX; rerun with -h for copyright info
==30833== Command: ./main.o
==30833==
Length of array: 1
84
==30833==
==30833== HEAP SUMMARY:
==30833==   in use at exit: 0 bytes in 0 blocks
==30833==   total heap usage: 2 allocs, 2 frees, 73,728 bytes allocated
==30833==
==30833== All heap blocks were freed -- no leaks are possible
==30833==
==30833== For lists of detected and suppressed errors, rerun with: -s
==30833== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
root@aw313146638:~/labs/Labs_PSTU/Sem_2/Green/4# valgrind ./main.o
==30836== Memcheck, a memory error detector
==30836== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==30836== Using Valgrind-3.16.1 and LibVEX; rerun with -h for copyright info
==30836== Command: ./main.o
==30836==
Length of array: 9
31 -5 -84 -17 -78 80 -52 52 31
-5 4 -84 83 -78 77 80 52 31
==30836==
==30836== HEAP SUMMARY:
==30836==   in use at exit: 0 bytes in 0 blocks
==30836==   total heap usage: 2 allocs, 2 frees, 73,728 bytes allocated
==30836==
==30836== All heap blocks were freed -- no leaks are possible
==30836==
==30836== For lists of detected and suppressed errors, rerun with: -s
==30836== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
root@aw313146638:~/labs/Labs_PSTU/Sem_2/Green/4#
```

Рис. 1: Сборка и запуск программы main.o с помощью valgrind