Пермский национальный исследовательский политехнический университет

Электротехнический факультет

Информатика и вычислительная техника

ОТЧЕТ

О выполнении лабораторной работы №8
«Блоковый ввод-вывод»

Студент:	Шадрин И. Д.
Группа:	ИВТ-23-1Б
Преподаватель:	Яруллин Д. В.

Пермь – 2024

1. Формулировка индивидуального задания

Сформировать двоичный файл из элементов, заданной в варианте структуры, распечатать его содержимое, выполнить удаление всех элементов с заданным возрастом, добавить элемент после элемента с заданным номером.

Структура “Человек” :

- Фамилия, Имя, Отчество;
- Домашний адрес;
- Номер телефона;
- Возраст.

2. Описание использованных типов данных

При выполнении данной лабораторной работы использовался встроенный тип данных int, предназначенный для работы с целыми числами, шаблон vector и библиотечный класс string.

3. Исходные коды разработанных программ

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

```
#include <iostream>
#include <vector>
#include <string>
#include "person.h"

void write_to_bin_file(const std::vector<Person*>& arr, std::string filename);
void read_from_bin_file(std::vector<Person*>& arr, std::string filename);
void delete_arr(std::vector<Person*>& arr);
void delete_by_age(std::vector<Person*>& arr);
void add_by_index(std::vector<Person*>& arr);
void print_arr(const std::vector<Person*>& arr);

int main() {
    std::vector<Person*> arr;
    std::string filename;
    std::cout << "Enter filename: ";
    std::getline(std::cin, filename);

    char ans = 'y';
    while (ans != 'n') {
        Person *p = create_person();
        std::string name, address, phone;
        int age;

        std::cout << "Enter name: ";
        std::getline(std::cin, name);
        std::cout << "Enter address: ";
        std::getline(std::cin, address);
        std::cout << "Enter phone: ";
        std::getline(std::cin, phone);
        std::cout << "Enter age: ";
        std::cin >> age;
        std::cin.ignore();

        add_data_to_person(p, name, address, phone, age);
        arr.push_back(p);

        std::cout << "Do you want to add another person? (y/n): ";
        std::cin >> ans;
        std::cin.ignore();
    }

    write_to_bin_file(arr, filename);
    delete_arr(arr);

    read_from_bin_file(arr, filename);
    print_arr(arr);
    delete_by_age(arr);
    add_by_index(arr);
    write_to_bin_file(arr, filename);
    delete_arr(arr);

    read_from_bin_file(arr, filename);
    print_arr(arr);
    delete_arr(arr);

    return 0;
}

void write_to_bin_file(const std::vector<Person*>& arr, std::string filename) {
    FILE *file = fopen(filename.c_str(), "wb");

    int size = arr.size();
    fwrite(&size, sizeof(int), 1, file);

    for (Person* person : arr) {
        int tmp;

        tmp = get_name(person).length();
        fwrite(&tmp, sizeof(int), 1, file);
        fwrite(get_name(person).c_str(), sizeof(char), tmp, file);

        tmp = get_address(person).length();
        fwrite(&tmp, sizeof(int), 1, file);
        fwrite(get_address(person).c_str(), sizeof(char), tmp, file);

        tmp = get_phone(person).length();
        fwrite(&tmp, sizeof(int), 1, file);
        fwrite(get_phone(person).c_str(), sizeof(char), tmp, file);

        tmp = get_age(person);
        fwrite(&tmp, sizeof(int), 1, file);
    }

    fclose(file);
}

void read_from_bin_file(std::vector<Person*>& arr, std::string filename) {
    FILE *file = fopen(filename.c_str(), "rb");

    int size;
    fread(&size, sizeof(int), 1, file);

    for (int i = 0; i < size; i++) {
        Person *p = create_person();
        int tmp;

        fread(&tmp, sizeof(int), 1, file);
        std::string name;
        name.resize(tmp);
        fread(&name[0], sizeof(char), tmp, file);

        fread(&tmp, sizeof(int), 1, file);
        std::string address;
        address.resize(tmp);
        fread(&address[0], sizeof(char), tmp, file);

        fread(&tmp, sizeof(int), 1, file);
        std::string phone;
        phone.resize(tmp);
        fread(&phone[0], sizeof(char), tmp, file);

        fread(&tmp, sizeof(int), 1, file);

        add_data_to_person(p, name, address, phone, tmp);

        arr.push_back(p);
    }

    fclose(file);
}

void delete_arr(std::vector<Person*>& arr) {
    for (Person* person : arr) {
        delete_person(person);
    }

    arr.clear();
    arr.shrink_to_fit();
}

void delete_by_age(std::vector<Person*>& arr) {
    int age;
    std::cout << "Enter age to delete: ";
    std::cin >> age;

    auto it = arr.begin();
    while (it != arr.end()) {
        if (get_age(*it) == age) {
            delete_person(*it);
            it = arr.erase(it);
        } else {
            ++it;
        }
    }
}

void add_by_index(std::vector<Person*>& arr) {
    int index;
    std::cout << "Enter index to add: ";
    std::cin >> index;
```

```cpp
    if (index < 0 || index > arr.size()) {
        std::cout << "Invalid index" << std::endl;
        return;
    }

    Person *p = create_person();
    std::string name, address, phone;
    int age;

    std::cin.ignore();
    std::cout << "Enter name: ";
    std::getline(std::cin, name);
    std::cout << "Enter address: ";
    std::getline(std::cin, address);
    std::cout << "Enter phone: ";
    std::getline(std::cin, phone);
    std::cout << "Enter age: ";
    std::cin >> age;
    std::cin.ignore();

    add_data_to_person(p, name, address, phone, age);
    arr.insert(arr.begin() + index + 1, p);
}

void print_arr(const std::vector<Person*>& arr) {
    for (Person* person : arr) {
        std::cout << "Name: " << get_name(person) << std::endl;
        std::cout << "Address: " << get_address(person) << std::endl;
        std::cout << "Phone: " << get_phone(person) << std::endl;
        std::cout << "Age: " << get_age(person) << std::endl;
        std::cout << std::endl;
    }
}
```

Листинг 2: Исходные коды программы `main.o` (файл: `person.cpp`)

```
#include "person.h"

struct Person {
    std::string name;
    std::string address;
    std::string phone;
    int age;
};

Person *create_person() {
    return new Person();
}

void add_data_to_person(Person *person, std::string name, std::string adress,
std::string phone, int age) {
    person->name = name;
    person->address = adress;
    person->phone = phone;
    person->age = age;
}

std::string get_name(Person *person) {
    return person->name;
}

std::string get_address(Person *person) {
    return person->address;
}

std::string get_phone(Person *person) {
    return person->phone;
}

int get_age(Person *person) {
    return person->age;
}

void delete_person(Person *person) {
    delete person;
}
```

Листинг 3: Исходные коды программы `main.o` (файл: `person.h`)

```
#ifndef PERSON_H
#define PERSON_H

#include <iostream>

typedef struct Person Person;

Person *create_person();
void add_data_to_person(Person *person, std::string name, std::string adress,
std::string phone, int age);
std::string get_name(Person *person);
std::string get_address(Person *person);
std::string get_phone(Person *person);
int get_age(Person *person);
void delete_person(Person *person);

#endif
```

4. Описание тестовых примеров

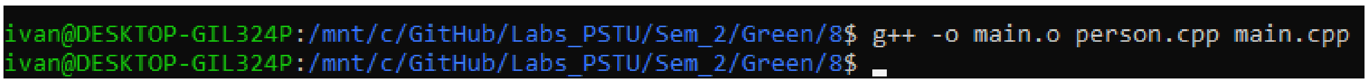
Таблица 1: Тестовые примеры

Ввод	Ожидаемый вывод	Вывод
Enter filename: test.bin Enter name: Test1 Test1 Test1 Enter address: Test11, 12 Enter phone: +79999999999 Enter age: 25 Do you want to add another person? (y/n): y Enter name: Test21 Test22 Test23 Enter address: Test2222, 12, 21 Enter phone: +78888888888 Enter age: 30 Do you want to add another person? (y/n): n Enter age to delete: 25 Enter index to add: 0 Enter name: Test31 Test32 Test33 Enter address: Test3333 Enter phone: +77777777777 Enter age: 35 Name: Test21 Test22 Test23 Address: Test2222, 12, 21 Phone: +78888888888 Age: 30	Name: Test1 Test1 Test1 Address: Test11, 12 Phone: +79999999999 Age: 25 Name: Test21 Test22 Test23 Address: Test2222, 12, 21 Phone: +78888888888 Age: 30 Name: Test21 Test22 Test23 Address: Test2222, 12, 21 Phone: +78888888888 Age: 30 Name: Test31 Test32 Test33 Address: Test3333 Phone: +77777777777 Age: 35	Name: Test1 Test1 Test1 Address: Test11, 12 Phone: +79999999999 Age: 25 Name: Test21 Test22 Test23 Address: Test2222, 12, 21 Phone: +78888888888 Age: 30 Name: Test21 Test22 Test23 Address: Test2222, 12, 21 Phone: +78888888888 Age: 30 Name: Test31 Test32 Test33 Address: Test3333 Phone: +77777777777 Age: 35
Enter filename: test.bin Enter name: Test1	Name: Test1 Address: Test2	Name: Test1 Address: Test2

Enter address: Test2 Enter phone: Test3 Enter age: 4 Do you want to add another person? (y/n): y Enter name: Test5 Enter address: Test6 Enter phone: Test7 Enter age: 8 Do you want to add another person? (y/n): y Enter name: Test9 Enter address: Test10 Enter phone: Test11 Enter age: 4 Do you want to add another person? (y/n): n Enter age to delete: 4 Enter index to add: 0 Enter name: Test13 Enter address: Test14 Enter phone: Test15 Enter age: 16	Phone: Test3 Age: 4 Name: Test5 Address: Test6 Phone: Test7 Age: 8 Name: Test9 Address: Test10 Phone: Test11 Age: 4 Name: Test5 Address: Test6 Phone: Test7 Age: 8 Name: Test13 Address: Test14 Phone: Test15 Age: 16	Phone: Test3 Age: 4 Name: Test5 Address: Test6 Phone: Test7 Age: 8 Name: Test9 Address: Test10 Phone: Test11 Age: 4 Name: Test5 Address: Test6 Phone: Test7 Age: 8 Name: Test13 Address: Test14 Phone: Test15 Age: 16
Enter filename: test.bin Enter name: test1 Enter address: test2 Enter phone: test3 Enter age: 4 Do you want to add another person? (y/n): n	Name: test1 Address: test2 Phone: test3 Age: 4 Name: test1 Address: test2	Name: test1 Address: test2 Phone: test3 Age: 4 Name: test1 Address: test2

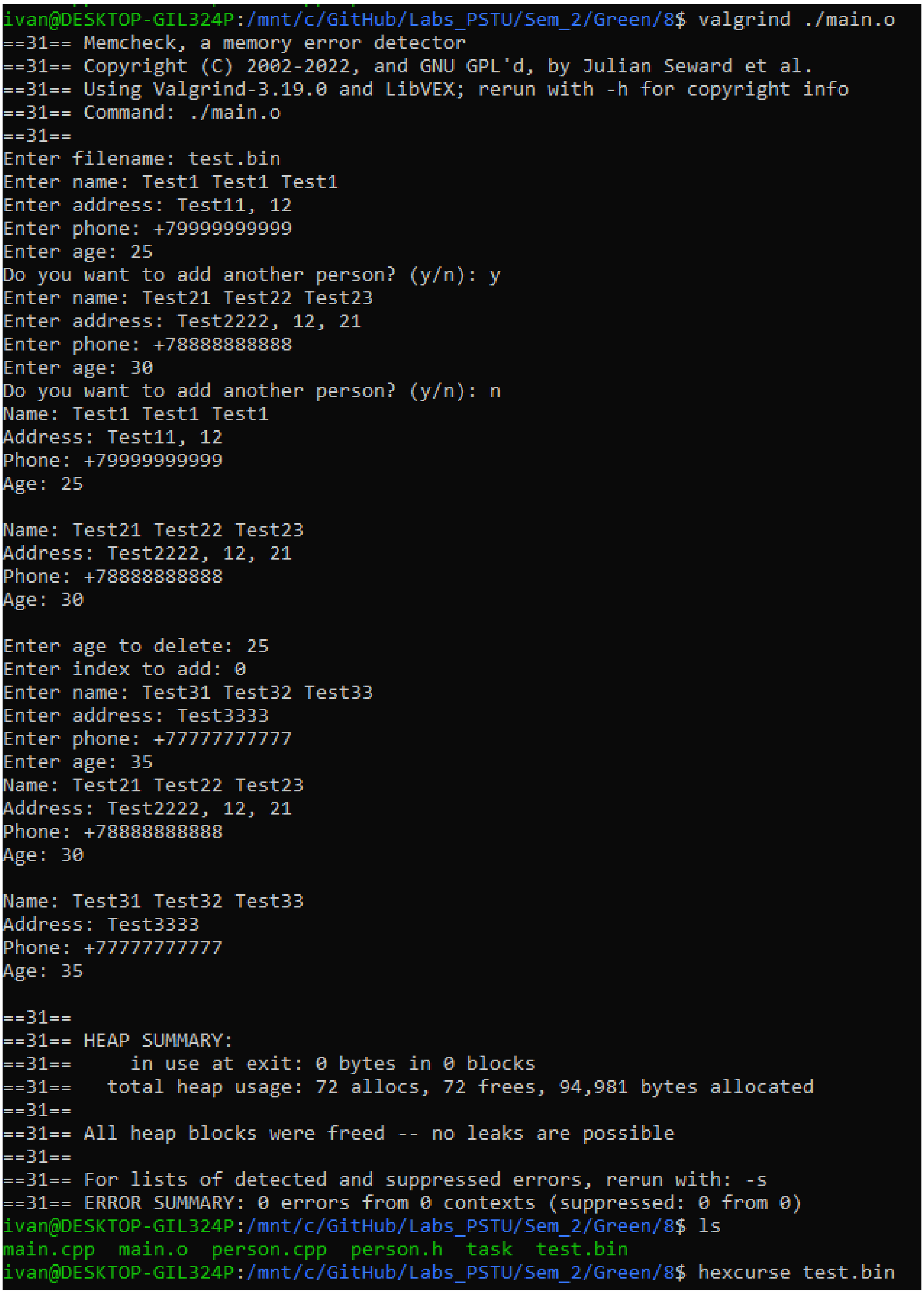
Enter age to delete: 5 Enter index to add: 0 Enter name: test2 Enter address: test3 Enter phone: test4	Phone: test3 Age: 4 Name: test2 Address: test3 Phone: test4 Age: 5	Phone: test3 Age: 4 Name: test2 Address: test3 Phone: test4 Age: 5

5. Скриншоты



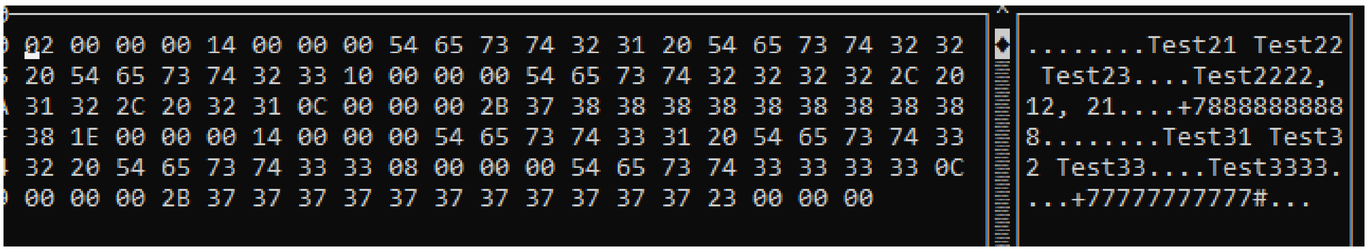
```
ivan@DESKTOP-GIL324P:/mnt/c/GitHub/Labs_PSTU/Sem_2/Green/8$ g++ -o main.o person.cpp main.cpp
ivan@DESKTOP-GIL324P:/mnt/c/GitHub/Labs_PSTU/Sem_2/Green/8$ _
```

Рис. 1: Сборка программы `main.o`



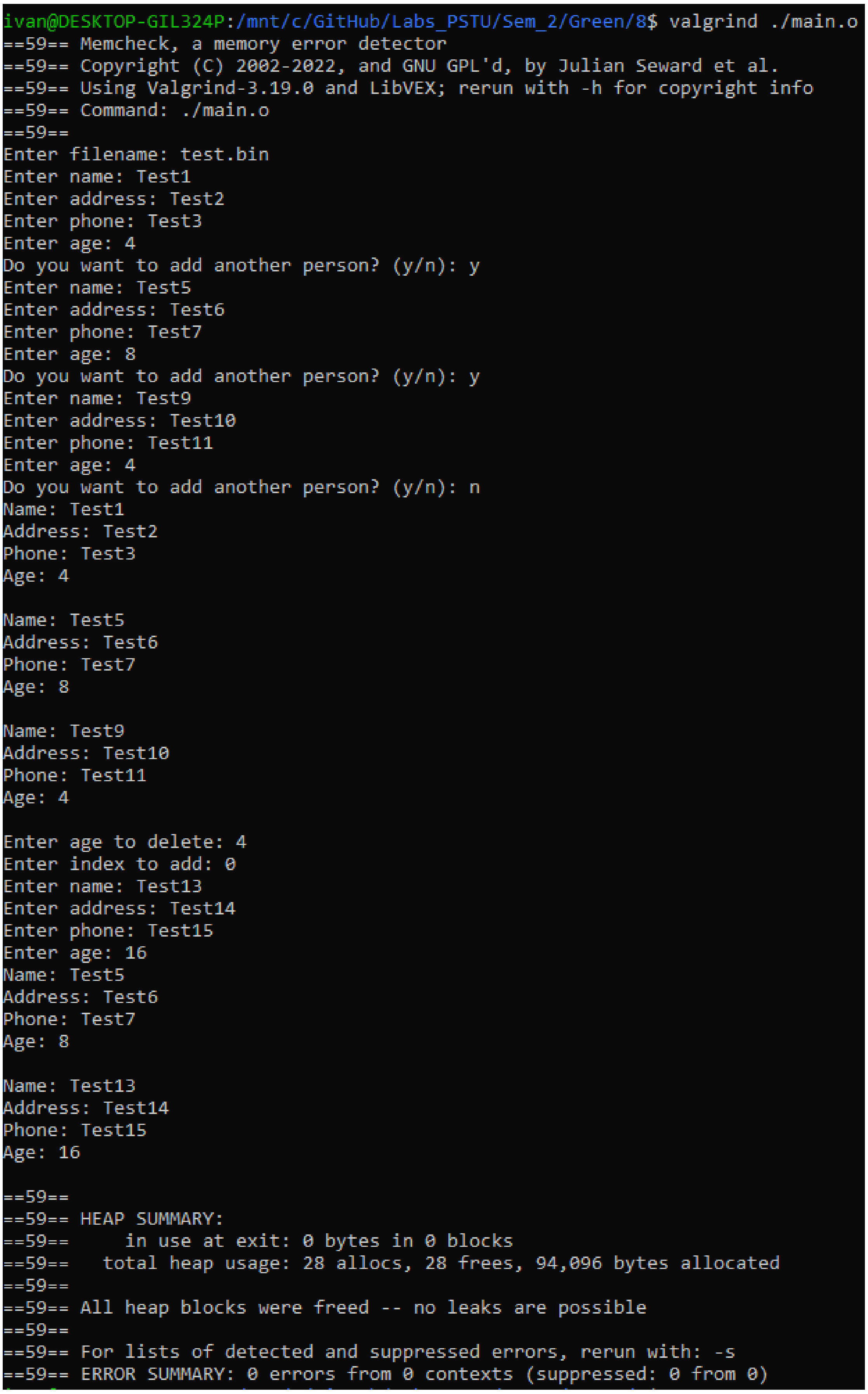
```
ivan@DESKTOP-GIL324P:/mnt/c/GitHub/Labs_PSTU/Sem_2/Green/8$ valgrind ./main.o
==31== Memcheck, a memory error detector
==31== Copyright (C) 2002-2022, and GNU GPL'd, by Julian Seward et al.
==31== Using Valgrind-3.19.0 and LibVEX; rerun with -h for copyright info
==31== Command: ./main.o
==31==
Enter filename: test.bin
Enter name: Test1 Test1 Test1
Enter address: Test11, 12
Enter phone: +79999999999
Enter age: 25
Do you want to add another person? (y/n): y
Enter name: Test21 Test22 Test23
Enter address: Test2222, 12, 21
Enter phone: +78888888888
Enter age: 30
Do you want to add another person? (y/n): n
Name: Test1 Test1 Test1
Address: Test11, 12
Phone: +79999999999
Age: 25

Name: Test21 Test22 Test23
Address: Test2222, 12, 21
Phone: +78888888888
Age: 30

Enter age to delete: 25
Enter index to add: 0
Enter name: Test31 Test32 Test33
Enter address: Test3333
Enter phone: +77777777777
Enter age: 35
Name: Test21 Test22 Test23
Address: Test2222, 12, 21
Phone: +78888888888
Age: 30

Name: Test31 Test32 Test33
Address: Test3333
Phone: +77777777777
Age: 35

==31==
==31== HEAP SUMMARY:
==31==     in use at exit: 0 bytes in 0 blocks
==31==   total heap usage: 72 allocs, 72 frees, 94,981 bytes allocated
==31==
==31== All heap blocks were freed -- no leaks are possible
==31==
==31== For lists of detected and suppressed errors, rerun with: -s
==31== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
ivan@DESKTOP-GIL324P:/mnt/c/GitHub/Labs_PSTU/Sem_2/Green/8$ ls
main.cpp  main.o  person.cpp  person.h  task  test.bin
ivan@DESKTOP-GIL324P:/mnt/c/GitHub/Labs_PSTU/Sem_2/Green/8$ hexcurse test.bin
```

Рис. 2: Запуск программы `main.o` с помощью `valgrind`



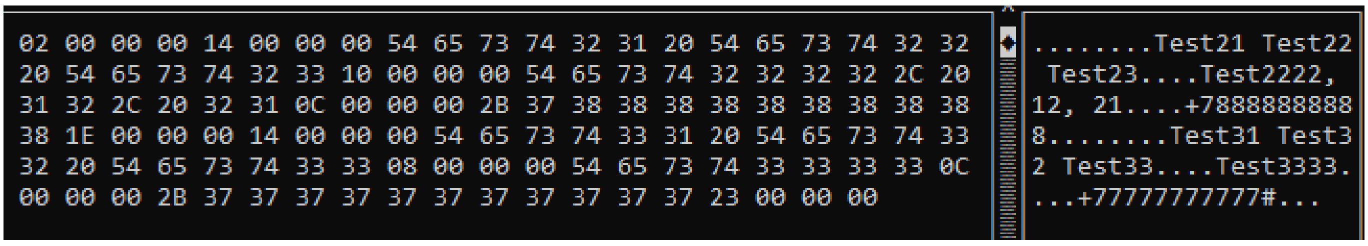
```
02 00 00 00 14 00 00 00 54 65 73 74 32 31 20 54 65 73 74 32 32   ........Test21 Test22
20 54 65 73 74 32 33 10 00 00 00 54 65 73 74 32 32 32 32 2C 20    Test23....Test2222,
31 32 2C 20 32 31 0C 00 00 00 2B 37 38 38 38 38 38 38 38 38 38   12, 21....+7888888888
38 1E 00 00 00 14 00 00 00 54 65 73 74 33 31 20 54 65 73 74 33   8........Test31 Test3
32 20 54 65 73 74 33 33 08 00 00 00 54 65 73 74 33 33 33 33 0C   2 Test33....Test3333.
00 00 00 2B 37 37 37 37 37 37 37 37 37 37 37 23 00 00 00         ...+77777777777#...
```

Рис. 3: Структура бинарного файла `test.bin`



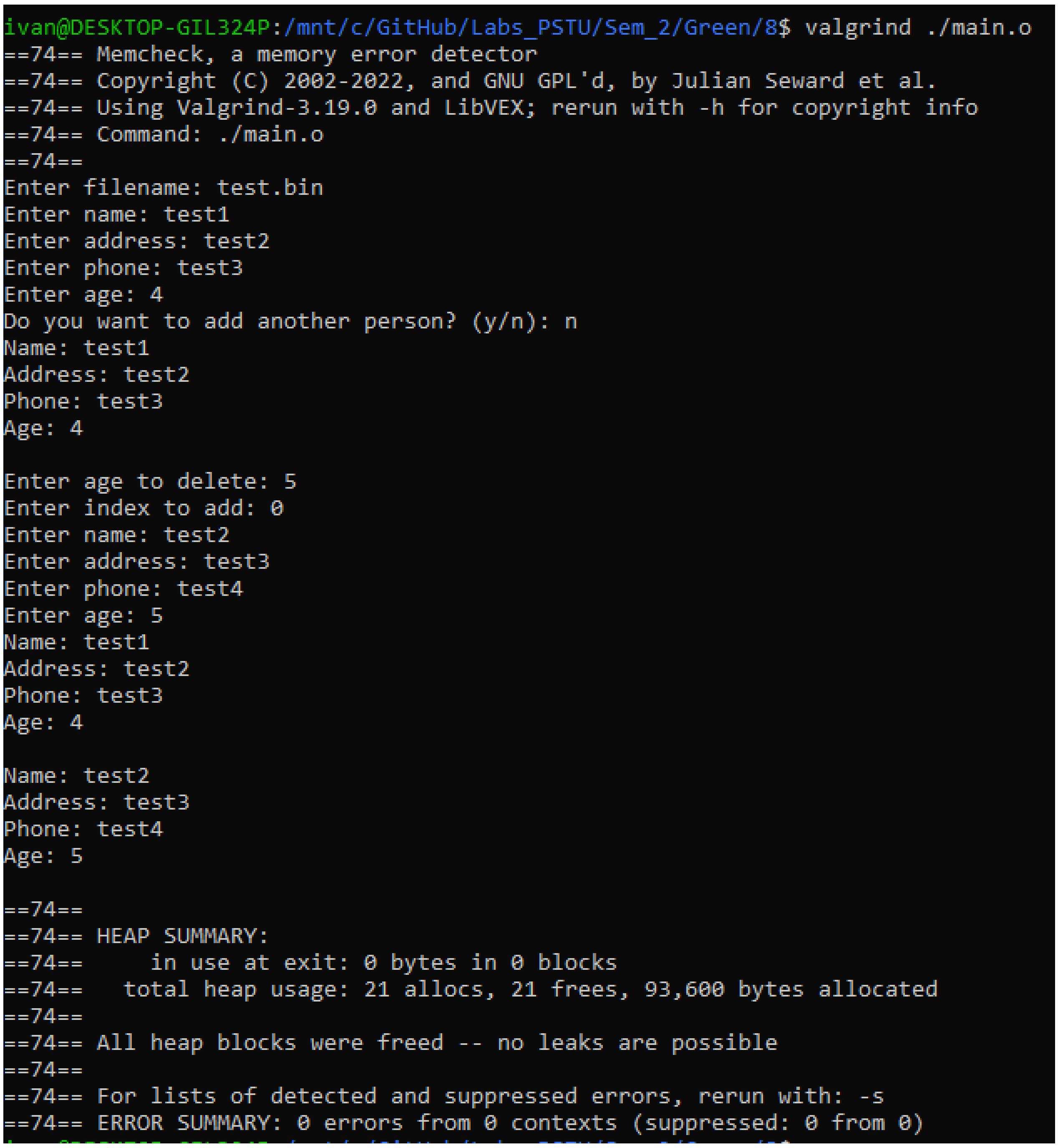
```
ivan@DESKTOP-GIL324P:/mnt/c/GitHub/Labs_PSTU/Sem_2/Green/8$ valgrind ./main.o
==59== Memcheck, a memory error detector
==59== Copyright (C) 2002-2022, and GNU GPL'd, by Julian Seward et al.
==59== Using Valgrind-3.19.0 and LibVEX; rerun with -h for copyright info
==59== Command: ./main.o
==59==
Enter filename: test.bin
Enter name: Test1
Enter address: Test2
Enter phone: Test3
Enter age: 4
Do you want to add another person? (y/n): y
Enter name: Test5
Enter address: Test6
Enter phone: Test7
Enter age: 8
Do you want to add another person? (y/n): y
Enter name: Test9
Enter address: Test10
Enter phone: Test11
Enter age: 4
Do you want to add another person? (y/n): n
Name: Test1
Address: Test2
Phone: Test3
Age: 4

Name: Test5
Address: Test6
Phone: Test7
Age: 8

Name: Test9
Address: Test10
Phone: Test11
Age: 4

Enter age to delete: 4
Enter index to add: 0
Enter name: Test13
Enter address: Test14
Enter phone: Test15
Enter age: 16
Name: Test5
Address: Test6
Phone: Test7
Age: 8

Name: Test13
Address: Test14
Phone: Test15
Age: 16

==59==
==59== HEAP SUMMARY:
==59==     in use at exit: 0 bytes in 0 blocks
==59==   total heap usage: 28 allocs, 28 frees, 94,096 bytes allocated
==59==
==59== All heap blocks were freed -- no leaks are possible
==59==
==59== For lists of detected and suppressed errors, rerun with: -s
==59== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

Рис. 4: Запуск программы `main.o` с помощью `valgrind`



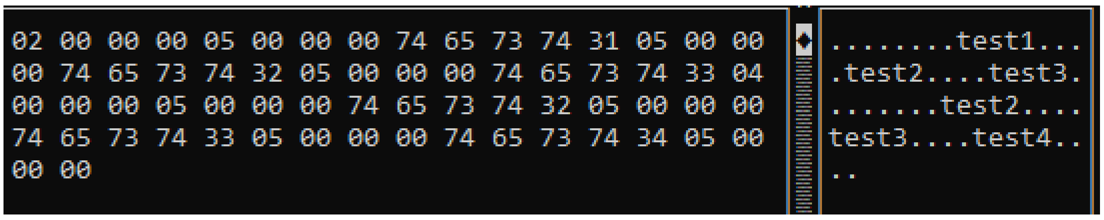
```
02 00 00 00 14 00 00 00 54 65 73 74 32 31 20 54 65 73 74 32 32   ........Test21 Test22
20 54 65 73 74 32 33 10 00 00 00 54 65 73 74 32 32 32 32 2C 20    Test23....Test2222,
31 32 2C 20 32 31 0C 00 00 00 2B 37 38 38 38 38 38 38 38 38 38   12, 21....+7888888888
38 1E 00 00 00 14 00 00 00 54 65 73 74 33 31 20 54 65 73 74 33   8........Test31 Test3
32 20 54 65 73 74 33 33 08 00 00 00 54 65 73 74 33 33 33 33 0C   2 Test33....Test3333.
00 00 00 2B 37 37 37 37 37 37 37 37 37 37 37 23 00 00 00         ...+77777777777#...
```

Рис. 5: Структура бинарного файла `test.bin`



```
ivan@DESKTOP-GIL324P:/mnt/c/GitHub/Labs_PSTU/Sem_2/Green/8$ valgrind ./main.o
==74== Memcheck, a memory error detector
==74== Copyright (C) 2002-2022, and GNU GPL'd, by Julian Seward et al.
==74== Using Valgrind-3.19.0 and LibVEX; rerun with -h for copyright info
==74== Command: ./main.o
==74==
Enter filename: test.bin
Enter name: test1
Enter address: test2
Enter phone: test3
Enter age: 4
Do you want to add another person? (y/n): n
Name: test1
Address: test2
Phone: test3
Age: 4

Enter age to delete: 5
Enter index to add: 0
Enter name: test2
Enter address: test3
Enter phone: test4
Enter age: 5
Name: test1
Address: test2
Phone: test3
Age: 4

Name: test2
Address: test3
Phone: test4
Age: 5

==74==
==74== HEAP SUMMARY:
==74==     in use at exit: 0 bytes in 0 blocks
==74==   total heap usage: 21 allocs, 21 frees, 93,600 bytes allocated
==74==
==74== All heap blocks were freed -- no leaks are possible
==74==
==74== For lists of detected and suppressed errors, rerun with: -s
==74== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

Рис. 6: Запуск программы `main.o` с помощью `valgrind`



```
02 00 00 00 05 00 00 00 74 65 73 74 31 05 00 00   ........test1...
00 74 65 73 74 32 05 00 00 00 74 65 73 74 33 04   .test2....test3.
00 00 00 05 00 00 00 74 65 73 74 32 05 00 00 00   .......test2....
74 65 73 74 33 05 00 00 00 74 65 73 74 34 05 00   test3....test4..
00 00                                             ..
```

Рис. 7: Структура бинарного файла `test.bin`