**Правительство Российской Федерации**

**Федеральное государственное автономное образовательное учреждение высшего профессионального образования   
"Национальный исследовательский университет   
"Высшая школа экономики"**

Московский институт электроники и математики Национального

исследовательского университета "Высшая школа экономики"

Департамент прикладной математики

**ОТЧЕТ**

**По лабораторной работе №6**

**Модификация одномерных динамических массивов**

**По курсу «Алгоритмизация и программирование»**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  | ФИО студента | | Номер группы | Дата |
| Кармаев Александр Андреевич | БПМ211 | 15.10.2021 |
|  |
|  |
|  |

**Москва – 2021 г.**

**ЗАДАНИЕ (вариант №13)**

*Размер динамического массива вводится пользователем на этапе выполнения. Тип*

*массива указан в задании. Элементы массива вводятся с клавиатуры. Написать функции*

*заполнения массива и вывода массива. Написать функцию модификации массива*

*указанных элементов. Вспомогательные массивы не использовать.*

****

**РЕШЕНИЕ**

#include <stdio.h>

#include <stdlib.h> // библиотека для работы с памятью для создания массива, неизвестного размера(динамическое выделение памяти)

void cheaked\_input1(char\* str, int\* p) { // функция присвоит необходимое значение переменной в основной функции, находящейся по адресу(2-ой аргумент)

char string[256];

printf("%s", str);

fgets(string, sizeof(string), stdin);

while (sscanf(string, "%i", p) != 1) {

printf("Incorrect input, try again!! %s", str);

fgets(string, sizeof(string), stdin);

}

}

void cheaked\_input2(char\* str1, int i, char\* str2, int\* p) { // перегрузка функций в СИ не работает, поэтому пришлось сделать так, потому что я не придумал, как можно создать одну строку из (строка + число + строка)

char string[256];

printf("%s%d%s", str1, i, str2);

fgets(string, sizeof(string), stdin);

while (sscanf(string, "%i", p) != 1) {

printf("Incorrect input, try again!! %s%d%s", str1, i, str2);

fgets(string, sizeof(string), stdin);

}

}

int return\_min(int B[], int k) { // функция возвращает индекс первого минимума массива

int m = 99999999999999999; // переменная, которой будет присвоено минимальное значение массива

int ind = 0; // переменная, которой будет присвоен индекс минимального значения массива

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

if (B[i] < m) {

m = B[i];

ind = i;

}

}

return ind;

}

void input\_in\_array(int B[], int k) { // функция ввода чисел в массив

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

cheaked\_input2("Input number B[", i, "] = ", &(B[i]));

}

void print\_array(char \*str, int B[], int k) { // функцию вывода элементов массива

printf("\nElements %s array: ", str);

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

printf("%d ", B[i]);

}

int\* modify\_array(int \*B, int k, int \*p) { // функция модифицирующая массив

int m = return\_min(B, k); // переменная хранящая значение индекса первого минимума массива

for (int i = m + 1; i < k; i++) { // пробегаем по всем элементам массива с минимального по последний

if (B[i] % 2 != 0) {

for (int j = i; j < k - 1; j++)

B[j] = B[j + 1]; // смещаем необходимый диапазон элементов влево и убираем первый(нечётный)(удаляем его из массива)

k--; // уменьшаем значение переменной, отвечающей за хранение размера массива(так как мы удалили один элемент)

B = realloc(B, k\*sizeof(int)); // функция вернёт указатель - начало памяти выделенной под наш массив, но меньшего размера, так как переменная k уменьшилась // мы перераспределили память под массив, убрали ненужную в связи с удалением элемента массива

print\_array("during modification", B, k); // выводим состояние массива после удаления элемента

i--; // уменьшаем текущий индекс массива, так как наш массив сдвинулся влево

}

}

\*p = k; // изменяем переменную из main(), отвечающую за хранение размера массива

return B;

}

int main(void) {

// объявляем переменную k - количество элементов массива

int k;

// записываем данные в k

do {

cheaked\_input1("Input number k in range[1, ...) = ", &k);

} while (k <= 0);

int\* B = malloc(k \* sizeof(int)); // создаём указатель(наш массив), которому присвоим указатель на начало выделенной под наш массив памяти // or calloc(k, sizeof(int));

input\_in\_array(B, k); // используем функцию ввода чисел в массив

print\_array("of the original", B, k); // используем функцию вывода элементов массива

printf("\n"); // делаем отступы для красоты

B = modify\_array(B, k, &k); // изменяем массив

printf("\n");

print\_array("of finally modified", B, k); // выводим новый массив

return 0;

}

**ТЕСТЫ**

**Тест № 1** // пример содержит все возможные обрабатываемые исключения

Input number k in range[1, ...) = -5

Input number k in range[1, ...) = 0

Input number k in range[1, ...) = 6

Input number B[0] = dd

Incorrect input, try again!! Input number B[0] = 5

Input number B[1] = 1

Input number B[2] = 2

Input number B[3] = 3

Input number B[4] = 4

Input number B[5] = 5

Elements of the original array: 5 1 2 3 4 5

Elements during modification array: 5 1 2 4 5

Elements of during modification array: 5 1 2 4

Elements of finally modified array: 5 1 2 4

**Тест № 2** // обычный пример

Input number k in range[1, ...) = 6

Input number B[0] = 5

Input number B[1] = 1

Input number B[2] = 2

Input number B[3] = 3

Input number B[4] = 5

Input number B[5] = 7

Elements of the original array: 5 1 2 3 5 7

Elements of during modification array: 5 1 2 5 7

Elements during modification array: 5 1 2 7

Elements of during modification array: 5 1 2

Elements of finally modified array: 5 1 2

**Тест № 3 //** пример с двумя минимума

Input number k in range[1, ...) = 10

Input number B[0] = 5

Input number B[1] = 1

Input number B[2] = 2

Input number B[3] = 3

Input number B[4] = 4

Input number B[5] = 5

Input number B[6] = 1

Input number B[7] = 2

Input number B[8] = 3

Input number B[9] = 4

Elements of the original array: 5 1 2 3 4 5 1 2 3 4

Elements during modification array: 5 1 2 4 5 1 2 3 4

Elements during modification array: 5 1 2 4 1 2 3 4

Elements during modification array: 5 1 2 4 2 3 4

Elements during modification array: 5 1 2 4 2 4

Elements of finally modified array: 5 1 2 4 2 4

**Тест № 4** // пример с отрицательными числами

Input number k in range[1, ...) = 6

Input number B[0] = -1

Input number B[1] = -5

Input number B[2] = -1

Input number B[3] = -2

Input number B[4] = -3

Input number B[5] = -4

Elements of the original array: -1 -5 -1 -2 -3 -4

Elements during modification array: -1 -5 -2 -3 -4

Elements during modification array: -1 -5 -2 -4

Elements of finally modified array: -1 -5 -2 -4

**Тест № 5** // пример с одним элементом массива

Input number k in range[1, ...) = 1

Input number B[0] = 1

Elements of the original array: 1

Elements of finally modified array: 1