Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт

Кафедра «Информатика»

кафедра

**ОТЧЕТ О ПРАКТИЧЕСКОЙ** **РАБОТЕ №1**

Работа с указателями (динамические массивы)

Тема

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Преподаватель |  |  |  |  |  | -- |
|  |  |  |  | подпись, дата |  | инициалы, фамилия |
| Студент |  | -- |  |  |  | -- |
|  |  | номер группы, зачетной книжки |  | подпись, дата |  | инициалы, фамилия |

Красноярск 2020

# СОДЕРЖАНИЕ

[1 Цель и задачи 3](#_Toc27678199)

[1.1 Цель 3](#_Toc27678200)

[1.2 Задачи 3](#_Toc27678201)

[2 Описание варианта задания 3](#_Toc27678202)

[2.1 Основное задание 3](#_Toc27678203)

[2.2 Дополнительное задание 4](#_Toc27678204)

[3 Ход выполнения 4](#_Toc27678205)

[4 Выводы 11](#_Toc27678206)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 12](#_Toc27678207)

# 1 Цель и задачи

## 1.1 Цель

Изучить принципы составления программной документации и оформления README.md файла на удаленном репозитории. Научиться писать документацию на Doxygen.

## 1.2 Задачи

На оценку 3 балла:

1. написать программу в соответствии с заданием, используя динамические массивы;
2. предусмотреть возможность ввода пользователем размера массива;
3. отформатировать исходный код программы согласно стандарту кодирования.

На оценку 4 балла:

1. выполнить все предыдущие пункты;
2. предусмотреть проверку вводимых пользователем значений и, если таковые имеются, предусмотреть вывод осмысленного сообщения;
3. организовать повтор программы по желанию пользователя.

На оценку 5 баллов:

1. выполнить все предыдущие пункты;
2. реализовать меню пользователя, состоящее минимум из 4-х пунктов (ввод элементов массива, обработка массива, вывод массива на экран, выход);
3. предусмотреть возможность заполнения массива случайными числами, либо числами, введенными пользователем с клавиатуры.

# 2 Описание варианта задания

Вариант №24. Дан массив чисел длиной n и число k. Необходимо сформировать новый массив такой, что его элементы будут равны номеру кластера, в который входит элемент соответствующего индекса из исходного массива. Числа входят в один кластер, если разность между ними меньше числа k.

# 3 Ход выполнения

Суть задания заключается в том, что необходимо распределить все числа на кластеры, другими словами, на небольшие группы, где в каждом кластере максимальное и минимальное числа не различаются больше чем на число k, которое вводит пользователь. Для удобства нумерация кластеров начинается от минимального числа в массиве.

Для начала мы, конечно, позволим пользователю ввести соответствующие числа n (длина массива), k (разность чисел в кластере) и выбор заполнить массив или вручную, или случайным образом.

Интерфейс программы поддерживает ввод массива, обработку массива, вывод массива или выход из программы по усмотрению пользователя. Реализация программы представлена ниже (листинг 1).

Листинг 1 – Реализованная программа с комментариями

#include <stdio.h>

#include <malloc.h>

#include <time.h>

int\* input(int\* n, int\* klu) {

// Function that catches user's input

// and transfers it to the determined variables

int \*arr;

int size = 0, k = 0;

int const area = 30;

// Asks the size of array

for (;;) {

if (size == 0) {

printf("How many elements will array be?\n");

scanf("%d", &size);

if (size <= 0) {

printf("Your array can't have negative elements\n");

size = 0;

continue;

}

}

break;

}

// Then what's the k - in-cluster difference.

Продолжение листинга 1.

for (;;) {

if (k == 0) {

printf("What is k?\n");

scanf("%d", &k);

if (k <= 0) {

printf("Make it positive\n");

k = 0;

continue;

}

}

break;

}

// Whether you want to fill the array manually or not.

int fill = 0;

printf("How do you want to fill the array?\n1. Random\n2. Manual\n");

for (;;) {

scanf("%d", &fill);

if (fill != 1 && fill != 2) {

printf("No such option\n");

continue;

}

break;

}

arr = calloc(size, sizeof(int));

srand(time(NULL));

// Actual array filling in.

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

if (fill == 2) {

printf("\nElement %d: ", i + 1);

scanf("%d", &arr[i]);

} else if (fill == 1) {

arr[i] = rand() % area;

}

}

// Returning values

\*klu = k;

\*n = size;

return arr;

}

int comp(const void \* elem1, const void \* elem2){

// Function to compare numbers for sorting

int f = \*((int\*)elem1);

int s = \*((int\*)elem2);

if (f > s) return 1;

if (f < s) return -1;

return 0;

}

static int\* calculate(int\* source, int const size, int const k){

// Actual calculations. It's assigns

// cluster numbers to the right elements

// init

int \*sorted, \*clusters = NULL;

sorted = calloc(size, sizeof(int));

Продолжение листинга 1.

clusters = calloc(size, sizeof(int));

int clust = 1;

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

clusters[i] = 0;

sorted[i] = source[i];

}

qsort(sorted, size, sizeof(int), comp);

// Actual comparing with k and assigning.

int fst = sorted[0];

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

if (sorted[0] == source[i]){

clusters[i] = 1;

}

}

for (int i = 1; i < size; i++){

if (sorted[i] == fst) continue;

else if (sorted[i] - fst < k){

;

}

else if (sorted[i] - fst >= k){

clust += 1;

fst = sorted[i];

}

for (int n = 0; n < size; n++) {

if (sorted[i] == source[n]) {

clusters[n] = clust;

}

}

}

// Freeing and returning

free(sorted);

return clusters;

}

int len(int n){

// Returns the length of a number

int l = 0, check = 0;

do{

n = n / 10;

if (n == 0) check = 1;

l++;

} while (check == 0);

return l;

}

void spaces(int am){

// "Important" function to print spaces

if (am > 1)

for (int i = 0; i < am; i++) printf(" ");

}

int main() {

// THE CORE

// Init

Продолжение листинга 1.

printf("Hello, World!\n");

int choice;

int \*src = NULL, \*result = NULL;

int n = 0, k = 0;

// User interface

for (;;) {

printf("\nWhatcha want?\n1. Enter array\n2. "

"Proceed array\n3. Output array\n4. Exit\n");

scanf("%d", &choice);

switch (choice) {

// Sends to input

case 1:

src = input(&n, &k);

free(result);

result = NULL;

break;

// Proceeds with calculation

case 2:

result = calculate(src, n, k);

break;

// Prints the result

case 3:

printf("[");

for (int loop = 0; loop < n; loop++) {

printf("%-3d ", src[loop]);

}

printf("] - source\n");

printf("[");

if (result == NULL) {

printf("Not proceeded yet");

} else {

for (int loop = 0; loop < n; loop++) {

printf("%-3d ", result[loop]);

spaces(len(src[loop]) - len(result[loop]));

}

}

printf("] - clusters\n");

break;

// Says goodbye

case 4: return 0;

// In case of incorrect option

default:

printf("There is no such option :c");

break;

}

}

}

# 4 Выводы

Задача была реализована на языке Си. Она способна распределить числа по кластерам в пронумерованном порядке и вывести соответственно начальный массив и массив с номерами кластеров. Для удобства реализована возможность случайного заполнения массива натуральными числами.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. СТО 4.2-07-2014 Система менеджмента качества Общие требования к построению, изложению и оформлению документов учебной деятельности. – Взамен СТО 4.2–07–2012 ; введ. 09.01.2014, - Красноярск : ИПК СФУ, 2014 – 60 с.