

**ОТЧЕТ**

по лабораторным работам

**МДК 01.01. Системное программирование**

по специальности 09.02.03 «Программирование в компьютерных системах»

Выполнили студенты гр. П1-17

Карандашев Н. К.

Юшаков Н. Р.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (подпись)

Проверил преподаватель

Гусятинер Л. Б.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(оценка)

Королев, 2019

Оглавление

[**Лабораторная работа №1** 3](#_Toc22671922)

[**Лабораторная работа №2** 4](#_Toc22671923)

[**Лабораторная работа №3** 7](#_Toc22671924)

[**Лабораторная работа №4** 10](#_Toc22671925)

[**Лабораторная работа №5** 13](#_Toc22671926)

[**Лабораторная работа №6** 18](#_Toc22671927)

[**Лабораторная работа №7** 23](#_Toc22671928)

[**Лабораторная работа №8** 28](#_Toc22671929)

[**Лабораторная работа №9** 31](#_Toc22671930)

[**Лабораторная работа №10** 36](#_Toc22671931)

[**Лабораторная работа №11** 38](#_Toc22671932)

[**Лабораторная работа №12** 40](#_Toc22671933)

[**Лабораторная работа №13** 45](#_Toc22671934)

[**Лабораторная работа №14** 46](#_Toc22671935)

[**Лабораторная работа №15** 47](#_Toc22671936)

[**Лабораторная работа №16** 49](#_Toc22671937)

[**Лабораторная работа №17** 50](#_Toc22671938)

[**Лабораторная работа №18** 51](#_Toc22671939)

[**Лабораторная работа №19** 53](#_Toc22671940)

[**Лабораторная работа №20** 54](#_Toc22671941)

[**Лабораторная работа №21** 55](#_Toc22671942)

[**Лабораторная работа №22** 56](#_Toc22671943)

[**Лабораторная работа №23** 57](#_Toc22671944)

[**Лабораторная работа №24** 59](#_Toc22671945)

[**Лабораторная работа №25** 61](#_Toc22671946)

[**Лабораторная работа №26** 62](#_Toc22671947)

[**Лабораторная работа №27** 64](#_Toc22671948)

[**Лабораторная работа №28** 65](#_Toc22671949)

# **Лабораторная работа №1**

**Цель:** Входной контроль

**Приложение:** Array22.py, Begin22.py, Boolean22.py, If22.py, Integer22.py, Minmax22.py, String22.py, While22.py

**Упражнение Boolean22:** Дано трехзначное число. Проверить истинность высказывания: «Цифры данного числа образуют возрастающую или убывающую последовательность».

///Выполнил Юшаков Н. Р.

a = int(input())

print(a // 100 > a % 100 // 10 > a % 10 or a // 100 < a % 100 // 10 < a % 10)

**Test input 1:** 146

**Test output 1:** True

**Test input 2:** 435

**Test output 2:** False

**Упражнение String22:** Дана строка, изображающая целое положительное число.

Вывести сумму цифр этого числа.

///Выполнил Карандашев Н. К.

a = input()

sm = 0

for i in a:

sm += int(i)

print(sm)

**Test input 1:** 146

**Test output 1:** 11

**Test input 2:** 435

**Test output 2:** 12

# **Лабораторная работа №2**

**Цель:** Разработка простейших программ на языке C

**Приложение:** 1.1.с, 1.2.с, 1.3.с, 1.4.с, 1.5.с, 1.6.с, 1.7.с, 1.8.с, 1.9.с, 1.10.с, 1.11.с, 1.12.с, 1.13.с, 1.14.с, 1.15.с, 1.20.c, 2.1.c, 4.14.c

**К.Р. Упражнение 1.12** Напишите программу для вывода входного потока по одному слову в строке.

#include <stdio.h>

/\*К. Р. Напишите программу для вывода входного потока по одному слову в строке \*/

/// Выполнил Юшаков Н. Р. 30.09.2019

int main()

{

int c;

while ((c = getchar()) != EOF)

if (c == ' ')

printf("\n");

else if (c == ',' || c == '.' || c == '!' || c == '?' || c == ':')

printf("%c\n", c);

else

putchar(c);

return 0;

}

**Test input 1:** hi world!

**Test output 1:** hi

world!

**Test input 2:** hi

**Test output 2:** hi

**К. Р. Упражнение 1.10** Напишите программу для копирования входного потока в выходной с заменой знаков табуляции на \t, символов возврата назад (Backspace) на \b, а обратных косых черт – на \\. Это сделает табуляции и символы возврата легко читаемыми в потоке.

#include <stdio.h>

///Выполнил Карандашев Н. К.

int main()

{

int c;

while ((c = getchar()) != EOF)

{

if(c == '\t')

printf("\\t");

else if (c == '\b')

printf("\\b");

else if (c == '\\')

printf("\\\\");

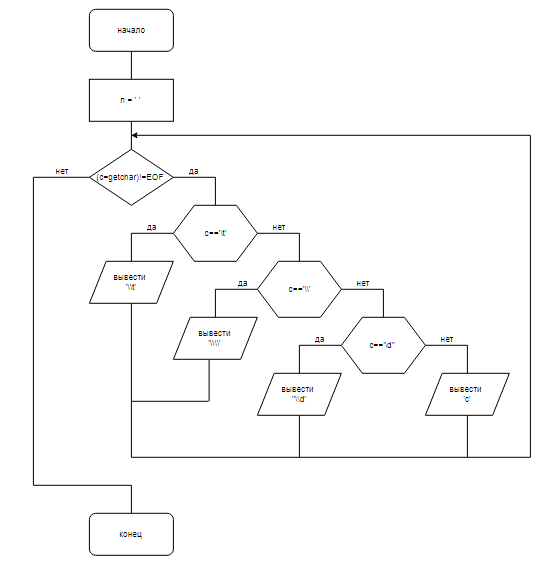
else

printf("%c", c);

}

return 0;

}



**Test input 1:** abc def

**Test output 1:** abc\tdef

**Test input 2:** abc \def

**Test output 2:** abc\t\\def

**К. Р. Упражнение 1.5** Доработайте программу преобразования температур так, чтобы она выводила таблицу в обратном порядке, т.е. от 300 градусов до нуля.

#include <stdio.h>

/\* Упражнение 1.5.

Доработайте программу преобразования температур так, чтобы она выводила таблицу в обратном порядке, т.е. от 300 градусов до нуля. \*/

///Выполнили Юшаков и Карандашев

int main()

{

int fahr;

for(fahr = 300; fahr >= 0; fahr -= 20)

printf("%3d %6.1lf\n", fahr, (5.0 / 9.0) \* (fahr - 32));

return 0;

}

# **Лабораторная работа №3**

**Цель:** Работа с массивами.

**Приложение:** 1.16.c, 1.17.c, 1.18.c, 1.19.c, 2.2.c, 2.3.c, 2.4.c, 2.5.c. 3.1.1.c 3.2.с 3.3.c 3.4.c 3.5.c 3.6.c 4.1.c 4.2.c 4.7.c 4.8.с 4.10.с 4.11.c 4.12.c 4.13.c

**К. Р. Упражнение 1.19** Напишите функцию reverse(s), которая переписывает свой строковый аргумент s в обратном порядке. Воспользуйтесь ею для написания программы, которая бы выполняла такое обращение над каждой строкой входного потока по очереди.

#include <stdio.h>

#define MAXLINE 1000

///Упражнение 1.19. Напишите функцию reverse(s), которая переписывает свой строковый аргумент s в обратном порядке.

///Воспользуйтесь ею для написания программы, которая бы выполняла такое обращение над каждой строкой входного потока по очереди.

///Выполнил Юшаков Н. Р.

int getline(char s[], int lim) ///Функция ввода строки

{

int c, i;

for (i = 0; i < lim - 1 && (c = getchar()) != EOF && c != '\n'; i++)

s[i] = c;

if (c = '\n')

{

s[i] = c;

i++;

}

s[i] = '\0';

return i;

}

void reverse(char s[])///Функция переписывания в обратном порядке

{

int i = 0, j = 0, temp, len;

while (s[i] != '\n')

i++;

i--;

len = i;

for (j = 0; j <= len / 2; j++)

{

temp = s[j];

s[j] = s[i];

s[i] = temp;

i--;

}

}

int main()

{

int len;

char line[MAXLINE];

while (getline(line, MAXLINE) > 1)

{

reverse(line);

printf("%s", line);

}

return 0;

}

**Test input 1:** 1542a

**Test output 1:** a2451

**Test input 2:** hi world

**Test output 2:** dlrow ih

**К. Р. Упражнение 2.2** Напишите цикл, эквивалентный приведенному выше циклу for, не используя операции && и ||. стр.56 for (i=0; i<lim-1 && (c=getchar()) != ' \n' && с != EOF; ++i) s[i] = c;

#include <stdio.h>

#define MAXLEN 1000

///Выполнил Карандашев Н. К.

int getline(char str[], int lim)

{

int c, i = 0;

while (i < lim - 1)

{

c = getchar();

if (c != '\n')

{

if (c != EOF)

str[i] = c;

else

break;

}

else

break;

i++;

}

if (c == '\n')

{

str[i] = c;

i++;

}

str[i] = '\0';

return i;

}

int main()

{

char str[MAXLEN + 1];

int len;

while ((len = getline(str, MAXLEN)) > 0)

{

printf("%d ", len);

printf("%s", str);

}

return 0;

}

**Test input 1:** hi world

**Test output 1:** 9 hi world

**Test input 2:** abcd

**Test output 2:** 5 abcd

# **Лабораторная работа №4**

**Цель:** Работа с указателями

**Приложение:** 5.1.c 5.2.c 5.3.c 5.4.c 5.4.c 5.5.c 5.6.c 5.7.c 5.8.c 5.9.c

**К. Р. Упражнение 5.3** Напишите свою версию функции strcat, продемонстрированной в главе 2, с применением указателей. Напоминаем, что strcat(s,t) копирует строку t в конец строки s.

Юшаков Н. Р.

#include <stdio.h>

#define MAXLEN 1000

int getline(char s[], int lim);

void strcat(char \*s, char \*t);

int main()

{

int len;

char s[MAXLEN], t[MAXLEN] = "hi";

while ((len = getline(s, MAXLEN)) > 0)

{

strcat(s, t);

puts(s);

}

return 0;

}

int getline(char s[], int lim)

{

int c, i = 0;

while (--lim > 0 && (c = getchar()) != EOF && c != '\n')

s[i++] = c;

if (c == '\n')

s[i++] = c;

s[i] = '\0';

return i;

}

void strcat(char \*s, char \*t)

{

while (\*s != '\n')

s++;

while (\*s++ = \*t++)

;

}

**Test input** hello\_

**Test output** hello\_hi

**К. Р. Упражнение 5.4** Напишите функцию strend(s,t), которая бы возвращала 1, если строка t присутствует в конце строки s, и 0 в противном случае.

Карандашев Н. К.

#include <stdio.h>

#define MAXLEN 1000

int getline(char s[], int lim);

int strend(char \*s, char \*t);

int main()

{

int len;

char s[MAXLEN], t[MAXLEN] = "nik\n";

while ((len = getline(s, MAXLEN)) > 0)

{

printf("%d\n", strend(s, t));

}

return 0;

}

int getline(char s[], int lim)

{

int c, i = 0;

while (--lim > 0 && (c = getchar()) != EOF && c != '\n')

s[i++] = c;

if (c == '\n')

s[i++] = c;

s[i] = '\0';

return i;

}

int strend(char \*s, char \*t)

{

char \*pt = t;

char \*ps = s;

while (\*s)

s++;

while (\*t)

t++;

while (\*s-- == \*t--)

if (t == pt || s == ps)

break;

if (\*s == \*t && t == pt && \*s != '\0')

return 1;

else

return 0;

}

**Test input** hi nik

**Test output** 1

**Test intput** hi

**Test output** 0

# **Лабораторная работа №5**

**Цель:** Работа с динамическими объектами

**Приложение:** Country.cbp main.c Prototypes.h Functions.c Country.txt Флаги Столицы Гимны

Юшаков Н. Р.

main.c

#include <stdio.h>

#include <locale.h>

#include "Prototypes.h"

int main()

{

setlocale(LC\_ALL, "");

int opeation = 1;

char b[MAXLEN];

read();

while (opeation != 0)

{

printf("Выберите операцию\n");

printf("1 : Добавление информации о новой стране в файл\n");

printf("2 : Определение столицы по названию государства\n");

printf("3 : Определение названия государства по городу-столице\n");

printf("4 : Вывод всей информации о заданном государстве\n");

printf("5 : Вывод названий всех стран, расположенных в заданной части света\n");

printf("6 : Расчёт и вывод плотности населения (в тыс. чел. на 1 кв. км) для каждого государства\n");

printf("7 : Вывод количества государств, расположенных в заданной части света\n");

printf("8 : Расчет общей площади государств, расположенных в заданной части света\n");

printf("9 : Показать флаг страны по ее названию\n");

printf("10 : Показать столицу государства по его названию\n");

printf("11 : Включить гимн укащанной страны\n");

printf("0 : Выход из программы\n");

printf("Примечание: Все названия вводить с большой буквы на английском языке, все абривиатуры по типу United Kingdoms писать как UK\n");

scanf("%d", &opeation);

switch (opeation)

{

case 1:

add();

break;

case 2:

printf("Введите название государства ");

scanf("%s", b);

capital(b);

break;

case 3:

printf("Введите город-столицу ");

scanf("%s", b);

country(b);

break;

case 4:

printf("Введите название государства, информаию о котором хотите получить ");

scanf("%s", b);

write(b);

break;

case 5:

printf("Введите часть света ");

scanf("%s", b);

allcountry(b);

break;

case 6:

density();

break;

case 7:

printf("Введите часть света ");

scanf("%s", b);

countcountry(b);

break;

case 8:

printf("Введите часть света ");

scanf("%s", b);

allarea(b);

break;

case 9:

printf("Введите название страны\n");

scanf("%s", b);

showflag(b);

break;

case 10:

printf("Введите название страны\n");

scanf("%s", b);

showcapital(b);

break;

case 11:

printf("Введите название страны\n");

scanf("%s", b);

playgimn(b);

break;

case 0:

opeation = 0;

break;

default:

printf("Неизвестный номер операции : %d\n", opeation);

break;

}

getch();

system("cls");

}

return 0;

}

Карандашев Н. К.

Prototypes.h

#ifndef PROTOTYPES\_H\_INCLUDED

#define PROTOTYPES\_H\_INCLUDED

#define MAXLEN 100

struct country

{

char name[MAXLEN]; ///Название страны

char capital[MAXLEN]; ///Название столицы

char side[MAXLEN]; ///В какой стороне света находится

long people; ///Количество людей

double area; ///Площадь

};

void read(); ///Считывание из файла уже имеющиеся данные

void add(); ///Запись новых данных в файл

void capital(char name[]); ///Вывод столицы по названию страны

void country(char capital[]); ///Вывод страны по столице

void write(char name[]); ///Вывести инф - цию о конкретной стране

void allcountry(char side[]); ///Вывод всех страны в определенной части света

void density(); ///Вывод плотности населения во всех странах

void countcountry(char side[]); ///Вывод количества стран в определенной части света

void allarea(char side[]); ///Подсчет общей площади всех стран в определенной части света

void showflag(char country[]); ///Показать флаг страны

void playgimn(char country[]); ///Включить гимн страны

void showcapital(char country[]); ///Показать столицу страны

#endif

Юшаков Н. Р. и Карандашев Н. К.

Functions.c

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "Prototypes.h"

struct country \*a = 0; ///Выделение дин памяти для структуры

int size = 0; ///Количество имеющихся структур

void read() ///Считывание из файла уже имеющиеся данные

{

int ip = 0;

FILE \*f = fopen("Country.txt", "r");

if (feof(f))

printf("В файле нет данных");

else

while (!feof(f))

{

if (ip == size)

{

ip += 1;

a = (struct country\*)realloc(a, (ip \* 2) \* sizeof(struct country));

}

fscanf(f, "%s", a[size].name);

fscanf(f, "%s", a[size].capital);

fscanf(f, "%s", a[size].side);

fscanf(f, "%ld", &a[size].people);

fscanf(f, "%lf", &a[size++].area);

}

fclose(f);

}

void write(char name[]) ///Вывести инф - цию о конкретной стране

{

int j;

for (j = 0; (strcmp(a[j].name, name) != 0) && j < size; j++)

;

if (strcmp(a[j].name, name) == 0)

{

printf("%s ", a[j].name);

printf("%s ", a[j].capital);

printf("%s ", a[j].side);

printf("%ld ", a[j].people);

printf("%f\n", a[j].area);

}

else

printf("Страны %s нет в файле\n", name);

}

void add() ///Запись новых данных в файл

{

FILE \*f = fopen("Country.txt", "a");

a = (struct country\*)realloc(a, (size + 1) \* sizeof(struct country));

printf("Введите название страны ");

scanf("%s", a[size].name);

fprintf(f, "\n%s ", a[size].name);

printf("Введите название столицы ");

scanf("%s", a[size].capital);

fprintf(f, "%s ", a[size].capital);

printf("Введите сторону света ");

scanf("%s", a[size].side);

fprintf(f, "%s ", a[size].side);

printf("Введите численность людей ");

scanf("%ld", &a[size].people);

fprintf(f, "%ld ", a[size].people);

printf("Введите площадь страны ");

scanf("%lf", &a[size].area);

fprintf(f, "%f", a[size++].area);

fclose(f);

}

void capital(char name[]) ///Вывод столицы по названию страны

{

int j;

for (j = 0; (strcmp(a[j].name, name) != 0) && j < size; j++)

;

if (strcmp(a[j].name, name) == 0)

printf("Столица %s - %s\n", a[j].name, a[j].capital);

else

printf("Страны %s нет в файле\n", name);

}

void country(char capital[]) ///Вывод страны по столице

{

int j;

for (j = 0; (strcmp(a[j].capital, capital) != 0) && j < size; j++)

;

if (strcmp(a[j].capital, capital) == 0)

printf("%s столица %s\n", a[j].capital, a[j].name);

else

printf("Столицы %s нет в файле\n", capital);

}

void allcountry(char side[]) ///Вывод всех страны в определенной части света

{

int j;

for (j = 0; j < size; j++)

if (strcmp(a[j].side, side) == 0)

printf("%s\n", a[j].name);

}

void countcountry(char side[]) ///Вывод всех страны в определенной части света

{

int j, count = 0;

for (j = 0; j < size; j++)

if (strcmp(a[j].side, side) == 0)

count++;

printf("Количество стран в %s : %d\n", side, count);

}

void allarea(char side[]) ///Подсчет общей площади всех стран в определенной части света

{

int j;

double allarea = 0;

for (j = 0; j < size; j++)

if (strcmp(a[j].side, side) == 0)

allarea += a[j].area;

printf("Общая площадь государств в %s : %f\n", side, allarea);

}

void density() ///Вывод плотности населения во всех странах

{

int j;

for (j = 0; j < size; j++)

printf("Плотность населения (тыс. чел на 1 кв. км) в %s : %f\n", a[j].name, (a[j].people \* 1000000) / (a[j].area \* 1000));

}

void showflag(char country[]) ///Показать флаг страны

{

char s[MAXLEN] = "Флаги\\\\";

strcat(s, country);

strcat(s, ".jpg");

system(s);

}

void showcapital(char country[]) ///Показать столицу страны

{

char s[MAXLEN] = "Столицы\\\\";

strcat(s, country);

strcat(s, ".jpg");

system(s);

}

void playgimn(char county[]) ///Включить гимн страны

{

char s[MAXLEN] = "Гимны\\\\";

strcat(s, county);

strcat(s, ".mp3");

system(s);

}

# **Лабораторная работа №6**

**Цель:** Работа со структурами

**Приложение:** Country.cbp main.c Prototypes.h Functions.c Country.txt Флаги Столицы Гимны

Юшаков Н. Р.

main.c

#include <stdio.h>

#include <locale.h>

#include "Prototypes.h"

int main()

{

setlocale(LC\_ALL, "");

int opeation = 1;

char b[MAXLEN];

read();

while (opeation != 0)

{

printf("Выберите операцию\n");

printf("1 : Добавление информации о новой стране в файл\n");

printf("2 : Определение столицы по названию государства\n");

printf("3 : Определение названия государства по городу-столице\n");

printf("4 : Вывод всей информации о заданном государстве\n");

printf("5 : Вывод названий всех стран, расположенных в заданной части света\n");

printf("6 : Расчёт и вывод плотности населения (в тыс. чел. на 1 кв. км) для каждого государства\n");

printf("7 : Вывод количества государств, расположенных в заданной части света\n");

printf("8 : Расчет общей площади государств, расположенных в заданной части света\n");

printf("9 : Показать флаг страны по ее названию\n");

printf("10 : Показать столицу государства по его названию\n");

printf("11 : Включить гимн укащанной страны\n");

printf("0 : Выход из программы\n");

printf("Примечание: Все названия вводить с большой буквы на английском языке, все абривиатуры по типу United Kingdoms писать как UK\n");

scanf("%d", &opeation);

switch (opeation)

{

case 1:

add();

break;

case 2:

printf("Введите название государства ");

scanf("%s", b);

capital(b);

break;

case 3:

printf("Введите город-столицу ");

scanf("%s", b);

country(b);

break;

case 4:

printf("Введите название государства, информаию о котором хотите получить ");

scanf("%s", b);

write(b);

break;

case 5:

printf("Введите часть света ");

scanf("%s", b);

allcountry(b);

break;

case 6:

density();

break;

case 7:

printf("Введите часть света ");

scanf("%s", b);

countcountry(b);

break;

case 8:

printf("Введите часть света ");

scanf("%s", b);

allarea(b);

break;

case 9:

printf("Введите название страны\n");

scanf("%s", b);

showflag(b);

break;

case 10:

printf("Введите название страны\n");

scanf("%s", b);

showcapital(b);

break;

case 11:

printf("Введите название страны\n");

scanf("%s", b);

playgimn(b);

break;

case 0:

opeation = 0;

break;

default:

printf("Неизвестный номер операции : %d\n", opeation);

break;

}

getch();

system("cls");

}

return 0;

}

Карандашев Н. К.

Prototypes.h

#ifndef PROTOTYPES\_H\_INCLUDED

#define PROTOTYPES\_H\_INCLUDED

#define MAXLEN 100

struct country

{

char name[MAXLEN]; ///Название страны

char capital[MAXLEN]; ///Название столицы

char side[MAXLEN]; ///В какой стороне света находится

long people; ///Количество людей

double area; ///Площадь

};

void read(); ///Считывание из файла уже имеющиеся данные

void add(); ///Запись новых данных в файл

void capital(char name[]); ///Вывод столицы по названию страны

void country(char capital[]); ///Вывод страны по столице

void write(char name[]); ///Вывести инф - цию о конкретной стране

void allcountry(char side[]); ///Вывод всех страны в определенной части света

void density(); ///Вывод плотности населения во всех странах

void countcountry(char side[]); ///Вывод количества стран в определенной части света

void allarea(char side[]); ///Подсчет общей площади всех стран в определенной части света

void showflag(char country[]); ///Показать флаг страны

void playgimn(char country[]); ///Включить гимн страны

void showcapital(char country[]); ///Показать столицу страны

#endif

Юшаков Н. Р. и Карандашев Н. К.

Functions.c

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "Prototypes.h"

struct country \*a = 0; ///Выделение дин памяти для структуры

int size = 0; ///Количество имеющихся структур

void read() ///Считывание из файла уже имеющиеся данные

{

int ip = 0;

FILE \*f = fopen("Country.txt", "r");

if (feof(f))

printf("В файле нет данных");

else

while (!feof(f))

{

if (ip == size)

{

ip += 1;

a = (struct country\*)realloc(a, (ip \* 2) \* sizeof(struct country));

}

fscanf(f, "%s", a[size].name);

fscanf(f, "%s", a[size].capital);

fscanf(f, "%s", a[size].side);

fscanf(f, "%ld", &a[size].people);

fscanf(f, "%lf", &a[size++].area);

}

fclose(f);

}

void write(char name[]) ///Вывести инф - цию о конкретной стране

{

int j;

for (j = 0; (strcmp(a[j].name, name) != 0) && j < size; j++)

;

if (strcmp(a[j].name, name) == 0)

{

printf("%s ", a[j].name);

printf("%s ", a[j].capital);

printf("%s ", a[j].side);

printf("%ld ", a[j].people);

printf("%f\n", a[j].area);

}

else

printf("Страны %s нет в файле\n", name);

}

void add() ///Запись новых данных в файл

{

FILE \*f = fopen("Country.txt", "a");

a = (struct country\*)realloc(a, (size + 1) \* sizeof(struct country));

printf("Введите название страны ");

scanf("%s", a[size].name);

fprintf(f, "\n%s ", a[size].name);

printf("Введите название столицы ");

scanf("%s", a[size].capital);

fprintf(f, "%s ", a[size].capital);

printf("Введите сторону света ");

scanf("%s", a[size].side);

fprintf(f, "%s ", a[size].side);

printf("Введите численность людей ");

scanf("%ld", &a[size].people);

fprintf(f, "%ld ", a[size].people);

printf("Введите площадь страны ");

scanf("%lf", &a[size].area);

fprintf(f, "%f", a[size++].area);

fclose(f);

}

void capital(char name[]) ///Вывод столицы по названию страны

{

int j;

for (j = 0; (strcmp(a[j].name, name) != 0) && j < size; j++)

;

if (strcmp(a[j].name, name) == 0)

printf("Столица %s - %s\n", a[j].name, a[j].capital);

else

printf("Страны %s нет в файле\n", name);

}

void country(char capital[]) ///Вывод страны по столице

{

int j;

for (j = 0; (strcmp(a[j].capital, capital) != 0) && j < size; j++)

;

if (strcmp(a[j].capital, capital) == 0)

printf("%s столица %s\n", a[j].capital, a[j].name);

else

printf("Столицы %s нет в файле\n", capital);

}

void allcountry(char side[]) ///Вывод всех страны в определенной части света

{

int j;

for (j = 0; j < size; j++)

if (strcmp(a[j].side, side) == 0)

printf("%s\n", a[j].name);

}

void countcountry(char side[]) ///Вывод всех страны в определенной части света

{

int j, count = 0;

for (j = 0; j < size; j++)

if (strcmp(a[j].side, side) == 0)

count++;

printf("Количество стран в %s : %d\n", side, count);

}

void allarea(char side[]) ///Подсчет общей площади всех стран в определенной части света

{

int j;

double allarea = 0;

for (j = 0; j < size; j++)

if (strcmp(a[j].side, side) == 0)

allarea += a[j].area;

printf("Общая площадь государств в %s : %f\n", side, allarea);

}

void density() ///Вывод плотности населения во всех странах

{

int j;

for (j = 0; j < size; j++)

printf("Плотность населения (тыс. чел на 1 кв. км) в %s : %f\n", a[j].name, (a[j].people \* 1000000) / (a[j].area \* 1000));

}

void showflag(char country[]) ///Показать флаг страны

{

char s[MAXLEN] = "Флаги\\\\";

strcat(s, country);

strcat(s, ".jpg");

system(s);

}

void showcapital(char country[]) ///Показать столицу страны

{

char s[MAXLEN] = "Столицы\\\\";

strcat(s, country);

strcat(s, ".jpg");

system(s);

}

void playgimn(char county[]) ///Включить гимн страны

{

char s[MAXLEN] = "Гимны\\\\";

strcat(s, county);

strcat(s, ".mp3");

system(s);

}

# **Лабораторная работа №7**

**Цель:** Работа с проектами

**Приложение:** Country.cbp main.c Prototypes.h Functions.c Country.txt Флаги Столицы Гимны

Юшаков Н. Р.

main.c

#include <stdio.h>

#include <locale.h>

#include "Prototypes.h"

int main()

{

setlocale(LC\_ALL, "");

int opeation = 1;

char b[MAXLEN];

read();

while (opeation != 0)

{

printf("Выберите операцию\n");

printf("1 : Добавление информации о новой стране в файл\n");

printf("2 : Определение столицы по названию государства\n");

printf("3 : Определение названия государства по городу-столице\n");

printf("4 : Вывод всей информации о заданном государстве\n");

printf("5 : Вывод названий всех стран, расположенных в заданной части света\n");

printf("6 : Расчёт и вывод плотности населения (в тыс. чел. на 1 кв. км) для каждого государства\n");

printf("7 : Вывод количества государств, расположенных в заданной части света\n");

printf("8 : Расчет общей площади государств, расположенных в заданной части света\n");

printf("9 : Показать флаг страны по ее названию\n");

printf("10 : Показать столицу государства по его названию\n");

printf("11 : Включить гимн укащанной страны\n");

printf("0 : Выход из программы\n");

printf("Примечание: Все названия вводить с большой буквы на английском языке, все абривиатуры по типу United Kingdoms писать как UK\n");

scanf("%d", &opeation);

switch (opeation)

{

case 1:

add();

break;

case 2:

printf("Введите название государства ");

scanf("%s", b);

capital(b);

break;

case 3:

printf("Введите город-столицу ");

scanf("%s", b);

country(b);

break;

case 4:

printf("Введите название государства, информаию о котором хотите получить ");

scanf("%s", b);

write(b);

break;

case 5:

printf("Введите часть света ");

scanf("%s", b);

allcountry(b);

break;

case 6:

density();

break;

case 7:

printf("Введите часть света ");

scanf("%s", b);

countcountry(b);

break;

case 8:

printf("Введите часть света ");

scanf("%s", b);

allarea(b);

break;

case 9:

printf("Введите название страны\n");

scanf("%s", b);

showflag(b);

break;

case 10:

printf("Введите название страны\n");

scanf("%s", b);

showcapital(b);

break;

case 11:

printf("Введите название страны\n");

scanf("%s", b);

playgimn(b);

break;

case 0:

opeation = 0;

break;

default:

printf("Неизвестный номер операции : %d\n", opeation);

break;

}

getch();

system("cls");

}

return 0;

}

Карандашев Н. К.

Prototypes.h

#ifndef PROTOTYPES\_H\_INCLUDED

#define PROTOTYPES\_H\_INCLUDED

#define MAXLEN 100

struct country

{

char name[MAXLEN]; ///Название страны

char capital[MAXLEN]; ///Название столицы

char side[MAXLEN]; ///В какой стороне света находится

long people; ///Количество людей

double area; ///Площадь

};

void read(); ///Считывание из файла уже имеющиеся данные

void add(); ///Запись новых данных в файл

void capital(char name[]); ///Вывод столицы по названию страны

void country(char capital[]); ///Вывод страны по столице

void write(char name[]); ///Вывести инф - цию о конкретной стране

void allcountry(char side[]); ///Вывод всех страны в определенной части света

void density(); ///Вывод плотности населения во всех странах

void countcountry(char side[]); ///Вывод количества стран в определенной части света

void allarea(char side[]); ///Подсчет общей площади всех стран в определенной части света

void showflag(char country[]); ///Показать флаг страны

void playgimn(char country[]); ///Включить гимн страны

void showcapital(char country[]); ///Показать столицу страны

#endif

Юшаков Н. Р. и Карандашев Н. К.

Functions.c

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "Prototypes.h"

struct country \*a = 0; ///Выделение дин памяти для структуры

int size = 0; ///Количество имеющихся структур

void read() ///Считывание из файла уже имеющиеся данные

{

int ip = 0;

FILE \*f = fopen("Country.txt", "r");

if (feof(f))

printf("В файле нет данных");

else

while (!feof(f))

{

if (ip == size)

{

ip += 1;

a = (struct country\*)realloc(a, (ip \* 2) \* sizeof(struct country));

}

fscanf(f, "%s", a[size].name);

fscanf(f, "%s", a[size].capital);

fscanf(f, "%s", a[size].side);

fscanf(f, "%ld", &a[size].people);

fscanf(f, "%lf", &a[size++].area);

}

fclose(f);

}

void write(char name[]) ///Вывести инф - цию о конкретной стране

{

int j;

for (j = 0; (strcmp(a[j].name, name) != 0) && j < size; j++)

;

if (strcmp(a[j].name, name) == 0)

{

printf("%s ", a[j].name);

printf("%s ", a[j].capital);

printf("%s ", a[j].side);

printf("%ld ", a[j].people);

printf("%f\n", a[j].area);

}

else

printf("Страны %s нет в файле\n", name);

}

void add() ///Запись новых данных в файл

{

FILE \*f = fopen("Country.txt", "a");

a = (struct country\*)realloc(a, (size + 1) \* sizeof(struct country));

printf("Введите название страны ");

scanf("%s", a[size].name);

fprintf(f, "\n%s ", a[size].name);

printf("Введите название столицы ");

scanf("%s", a[size].capital);

fprintf(f, "%s ", a[size].capital);

printf("Введите сторону света ");

scanf("%s", a[size].side);

fprintf(f, "%s ", a[size].side);

printf("Введите численность людей ");

scanf("%ld", &a[size].people);

fprintf(f, "%ld ", a[size].people);

printf("Введите площадь страны ");

scanf("%lf", &a[size].area);

fprintf(f, "%f", a[size++].area);

fclose(f);

}

void capital(char name[]) ///Вывод столицы по названию страны

{

int j;

for (j = 0; (strcmp(a[j].name, name) != 0) && j < size; j++)

;

if (strcmp(a[j].name, name) == 0)

printf("Столица %s - %s\n", a[j].name, a[j].capital);

else

printf("Страны %s нет в файле\n", name);

}

void country(char capital[]) ///Вывод страны по столице

{

int j;

for (j = 0; (strcmp(a[j].capital, capital) != 0) && j < size; j++)

;

if (strcmp(a[j].capital, capital) == 0)

printf("%s столица %s\n", a[j].capital, a[j].name);

else

printf("Столицы %s нет в файле\n", capital);

}

void allcountry(char side[]) ///Вывод всех страны в определенной части света

{

int j;

for (j = 0; j < size; j++)

if (strcmp(a[j].side, side) == 0)

printf("%s\n", a[j].name);

}

void countcountry(char side[]) ///Вывод всех страны в определенной части света

{

int j, count = 0;

for (j = 0; j < size; j++)

if (strcmp(a[j].side, side) == 0)

count++;

printf("Количество стран в %s : %d\n", side, count);

}

void allarea(char side[]) ///Подсчет общей площади всех стран в определенной части света

{

int j;

double allarea = 0;

for (j = 0; j < size; j++)

if (strcmp(a[j].side, side) == 0)

allarea += a[j].area;

printf("Общая площадь государств в %s : %f\n", side, allarea);

}

void density() ///Вывод плотности населения во всех странах

{

int j;

for (j = 0; j < size; j++)

printf("Плотность населения (тыс. чел на 1 кв. км) в %s : %f\n", a[j].name, (a[j].people \* 1000000) / (a[j].area \* 1000));

}

void showflag(char country[]) ///Показать флаг страны

{

char s[MAXLEN] = "Флаги\\\\";

strcat(s, country);

strcat(s, ".jpg");

system(s);

}

void showcapital(char country[]) ///Показать столицу страны

{

char s[MAXLEN] = "Столицы\\\\";

strcat(s, country);

strcat(s, ".jpg");

system(s);

}

void playgimn(char county[]) ///Включить гимн страны

{

char s[MAXLEN] = "Гимны\\\\";

strcat(s, county);

strcat(s, ".mp3");

system(s);

}

# **Лабораторная работа №8**

**Цель:** Передача сложных объектов в функции

**Приложение:** 3.1.1.с 3.2.c 3.3.c 3.4.c 3.5.c 3.6.c 4.1.c 4.2.c 4.7.с

**К. Р. Упражнение 3.1** В нашем двоичном поиске каждый цикл содержит две проверки, тогда как достаточно было бы одной

(зато взамен их потребовалось бы больше снаружи цикла). Напишите версию функции с одной проверкой внутри цикла

Юшаков Н. Р.

#include <stdio.h>

#define MAXLEN 11

int binsearch(int x, int v[], int n)

{

int low, high, mid;

low = high = mid = 0;

high = n - 1;

while (low <= high && x != v[mid])

{

mid = (low + high) / 2;

if (x < v[mid])

high = mid - 1;

else

low = mid + 1;

}

if (x == v[mid])

return mid;

else

return -1;

}

int main()

{

int v[MAXLEN] = {10, 8, 3, 6, 5, 4, 7, 2, 9, 1, 15}, x, temp;

scanf("%d", &x);

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

for (int j = 0; j < MAXLEN; j++)

if (v[i] < v[j])

{

temp = v[i];

v[i] = v[j];

v[j] = temp;

}

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

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

printf("\n%d", binsearch(x, v, MAXLEN));

return 0;

}

**Test** **input** 10

**Test** **output** 9

**Test input** 122

**Test output** -1

**К. Р. Упражнение 3.2** Напишите функцию под именем escape(s,t), которая бы преобразовывала символы наподобие конца строки и табуляции в управляющие последовательности языка С,

такие как \n и \t, в процессе копирования строки t в строку s. Воспользуйтесь оператором switch.

Напишите функцию также и для противоположной операции - преобразования символических управляющих последовательностей в фактические управляющие символы

Карандашев Н. К.

#include <stdio.h>

#define MAXLEN 1000

void escape(char s[], char t[]);

void escapeback(char s[], char t[]);

int getline(char s[], int lim);

int main()

{

char s[MAXLEN], t[MAXLEN], str[MAXLEN];

int len;

while ((len = getline(t, MAXLEN)) > 0)

{

escape(s, t);

puts(s);

escapeback(str, s);

puts(str);

}

return 0;

}

int getline(char s[], int lim)

{

int c, i;

for (i = 0; i < lim - 1 && (c = getchar()) != EOF && c != '\n'; i++)

s[i] = c;

if (c == '\n')

{

s[i] = c;

i++;

}

s[i] = '\0';

return i;

}

void escape(char s[], char t[])

{

int i, j;

for (j = i = 0; t[j] != '\0'; j++)

{

switch(t[j])

{

case '\n':

s[i++] = '\\';

s[i++] = 'n';

break;

case '\t':

s[i++] = '\\';

s[i++] = 't';

break;

default:

s[i++] = t[j];

break;

}

}

s[i] = '\0';

}

void escapeback(char s[], char t[])

{

int i, j;

for (j = i = 0; t[j] != '\0'; j++)

{

if (t[j] != '\\')

s[i++] = t[j];

else

switch(t[++j])

{

case 'n':

s[i++] = '\n';

break;

case 't':

s[i++] = '\t';

break;

default:

s[i++] = t[--j];

s[i++] = t[++j];

break;

}

}

s[i] = '\0';

}

**Test** **input** 123 123

**Test** **output** 123\t\t123\n

123 123

# **Лабораторная работа №9**

**Цель:** Работа с файлами

**Приложение:** Country.cbp main.c Prototypes.h Functions.c Country.txt Флаги Столицы Гимны

Юшаков Н. Р.

main.c

#include <stdio.h>

#include <locale.h>

#include "Prototypes.h"

int main()

{

setlocale(LC\_ALL, "");

int opeation = 1;

char b[MAXLEN];

read();

while (opeation != 0)

{

printf("Выберите операцию\n");

printf("1 : Добавление информации о новой стране в файл\n");

printf("2 : Определение столицы по названию государства\n");

printf("3 : Определение названия государства по городу-столице\n");

printf("4 : Вывод всей информации о заданном государстве\n");

printf("5 : Вывод названий всех стран, расположенных в заданной части света\n");

printf("6 : Расчёт и вывод плотности населения (в тыс. чел. на 1 кв. км) для каждого государства\n");

printf("7 : Вывод количества государств, расположенных в заданной части света\n");

printf("8 : Расчет общей площади государств, расположенных в заданной части света\n");

printf("9 : Показать флаг страны по ее названию\n");

printf("10 : Показать столицу государства по его названию\n");

printf("11 : Включить гимн укащанной страны\n");

printf("0 : Выход из программы\n");

printf("Примечание: Все названия вводить с большой буквы на английском языке, все абривиатуры по типу United Kingdoms писать как UK\n");

scanf("%d", &opeation);

switch (opeation)

{

case 1:

add();

break;

case 2:

printf("Введите название государства ");

scanf("%s", b);

capital(b);

break;

case 3:

printf("Введите город-столицу ");

scanf("%s", b);

country(b);

break;

case 4:

printf("Введите название государства, информаию о котором хотите получить ");

scanf("%s", b);

write(b);

break;

case 5:

printf("Введите часть света ");

scanf("%s", b);

allcountry(b);

break;

case 6:

density();

break;

case 7:

printf("Введите часть света ");

scanf("%s", b);

countcountry(b);

break;

case 8:

printf("Введите часть света ");

scanf("%s", b);

allarea(b);

break;

case 9:

printf("Введите название страны\n");

scanf("%s", b);

showflag(b);

break;

case 10:

printf("Введите название страны\n");

scanf("%s", b);

showcapital(b);

break;

case 11:

printf("Введите название страны\n");

scanf("%s", b);

playgimn(b);

break;

case 0:

opeation = 0;

break;

default:

printf("Неизвестный номер операции : %d\n", opeation);

break;

}

getch();

system("cls");

}

return 0;

}

Карандашев Н. К.

Prototypes.h

#ifndef PROTOTYPES\_H\_INCLUDED

#define PROTOTYPES\_H\_INCLUDED

#define MAXLEN 100

struct country

{

char name[MAXLEN]; ///Название страны

char capital[MAXLEN]; ///Название столицы

char side[MAXLEN]; ///В какой стороне света находится

long people; ///Количество людей

double area; ///Площадь

};

void read(); ///Считывание из файла уже имеющиеся данные

void add(); ///Запись новых данных в файл

void capital(char name[]); ///Вывод столицы по названию страны

void country(char capital[]); ///Вывод страны по столице

void write(char name[]); ///Вывести инф - цию о конкретной стране

void allcountry(char side[]); ///Вывод всех страны в определенной части света

void density(); ///Вывод плотности населения во всех странах

void countcountry(char side[]); ///Вывод количества стран в определенной части света

void allarea(char side[]); ///Подсчет общей площади всех стран в определенной части света

void showflag(char country[]); ///Показать флаг страны

void playgimn(char country[]); ///Включить гимн страны

void showcapital(char country[]); ///Показать столицу страны

#endif

Юшаков Н. Р. и Карандашев Н. К.

Functions.c

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "Prototypes.h"

struct country \*a = 0; ///Выделение дин памяти для структуры

int size = 0; ///Количество имеющихся структур

void read() ///Считывание из файла уже имеющиеся данные

{

int ip = 0;

FILE \*f = fopen("Country.txt", "r");

if (feof(f))

printf("В файле нет данных");

else

while (!feof(f))

{

if (ip == size)

{

ip += 1;

a = (struct country\*)realloc(a, (ip \* 2) \* sizeof(struct country));

}

fscanf(f, "%s", a[size].name);

fscanf(f, "%s", a[size].capital);

fscanf(f, "%s", a[size].side);

fscanf(f, "%ld", &a[size].people);

fscanf(f, "%lf", &a[size++].area);

}

fclose(f);

}

void write(char name[]) ///Вывести инф - цию о конкретной стране

{

int j;

for (j = 0; (strcmp(a[j].name, name) != 0) && j < size; j++)

;

if (strcmp(a[j].name, name) == 0)

{

printf("%s ", a[j].name);

printf("%s ", a[j].capital);

printf("%s ", a[j].side);

printf("%ld ", a[j].people);

printf("%f\n", a[j].area);

}

else

printf("Страны %s нет в файле\n", name);

}

void add() ///Запись новых данных в файл

{

FILE \*f = fopen("Country.txt", "a");

a = (struct country\*)realloc(a, (size + 1) \* sizeof(struct country));

printf("Введите название страны ");

scanf("%s", a[size].name);

fprintf(f, "\n%s ", a[size].name);

printf("Введите название столицы ");

scanf("%s", a[size].capital);

fprintf(f, "%s ", a[size].capital);

printf("Введите сторону света ");

scanf("%s", a[size].side);

fprintf(f, "%s ", a[size].side);

printf("Введите численность людей ");

scanf("%ld", &a[size].people);

fprintf(f, "%ld ", a[size].people);

printf("Введите площадь страны ");

scanf("%lf", &a[size].area);

fprintf(f, "%f", a[size++].area);

fclose(f);

}

void capital(char name[]) ///Вывод столицы по названию страны

{

int j;

for (j = 0; (strcmp(a[j].name, name) != 0) && j < size; j++)

;

if (strcmp(a[j].name, name) == 0)

printf("Столица %s - %s\n", a[j].name, a[j].capital);

else

printf("Страны %s нет в файле\n", name);

}

void country(char capital[]) ///Вывод страны по столице

{

int j;

for (j = 0; (strcmp(a[j].capital, capital) != 0) && j < size; j++)

;

if (strcmp(a[j].capital, capital) == 0)

printf("%s столица %s\n", a[j].capital, a[j].name);

else

printf("Столицы %s нет в файле\n", capital);

}

void allcountry(char side[]) ///Вывод всех страны в определенной части света

{

int j;

for (j = 0; j < size; j++)

if (strcmp(a[j].side, side) == 0)

printf("%s\n", a[j].name);

}

void countcountry(char side[]) ///Вывод всех страны в определенной части света

{

int j, count = 0;

for (j = 0; j < size; j++)

if (strcmp(a[j].side, side) == 0)

count++;

printf("Количество стран в %s : %d\n", side, count);

}

void allarea(char side[]) ///Подсчет общей площади всех стран в определенной части света

{

int j;

double allarea = 0;

for (j = 0; j < size; j++)

if (strcmp(a[j].side, side) == 0)

allarea += a[j].area;

printf("Общая площадь государств в %s : %f\n", side, allarea);

}

void density() ///Вывод плотности населения во всех странах

{

int j;

for (j = 0; j < size; j++)

printf("Плотность населения (тыс. чел на 1 кв. км) в %s : %f\n", a[j].name, (a[j].people \* 1000000) / (a[j].area \* 1000));

}

void showflag(char country[]) ///Показать флаг страны

{

char s[MAXLEN] = "Флаги\\\\";

strcat(s, country);

strcat(s, ".jpg");

system(s);

}

void showcapital(char country[]) ///Показать столицу страны

{

char s[MAXLEN] = "Столицы\\\\";

strcat(s, country);

strcat(s, ".jpg");

system(s);

}

void playgimn(char county[]) ///Включить гимн страны

{

char s[MAXLEN] = "Гимны\\\\";

strcat(s, county);

strcat(s, ".mp3");

system(s);

}

# **Лабораторная работа №10**

**Цель:** Работа с двоичными файлами

**Приложение:** 2.6.с 2.7.с 2.8.с 2.9.с

**К. Р. Упражнение 2.6** Напишите функцию setbits(x,p,n,y) так, чтобы она возвращала аргумент х, в котором n битов, начиная с позиции р, равны n крайним правым битам аргумента у, а остальные биты не тронуты.

Юшаков Н. Р.

int setbits(int x, int p, int n, int y)

{

int x1;

y = (y ^ (y >> n << n)) << (p + 1 - n);

x1 = x ^ (x >> (p + 1 - n) << (p + 1 - n));

x = (x >> n << n | x1) | y;

return x;

}

int main()

{

int a, b, n, p;

scanf("%d %d %d %d", &a, &p, &n, &b);

printf("%d", setbits(a, p, n, b));

return 0;

}

**Test input** 41 5 4 89

**Test output** 37

**К. Р. Упражнение 2.7** Напишите функцию invert(x,p,n), возвращающую свой аргумент х, в котором обращены п бит,

начиная с позиции р (т.е. единичные биты заменены нулевыми и наоборот), а остальные не тронуты.

Карандашев Н. К.

int invert(int x, int p, int n)

{

return x ^ ((~(~0 << n)) << (p + 1 - n));

}

int main()

{

int a;

scanf("%d", &a);

printf("%d", invert(a, 5, 4));

return 0;

}

**Test input** 41 5

**Test output** 21

# **Лабораторная работа №11**

**Цель**: Работа с библиотекой string.c

**Приложение**: 2.10.с ‘4.3 4.4 4.5 4.5 4.9.c’ 5.5.c

**К. Р. Упражнение 2.10** Перепишите функцию lower,

которая преобразует буквы в верхнем регистре к нижнему, с использованием условного выражения вместо конструкции if- else.

Юшаков Н. Р.

int lower(char c)

{

return (c >= 'A' && c <= 'Z') ? c + 'a' - 'A' : c;

}

int main()

{

char c;

while((c = getchar()) != EOF)

printf("%c", lower(c));

return 0;

}

**Test** **input** A B C D e F

**Test** **output** a b c d e f

**К. Р. Упражнение 5.5** Применяя указатели, напишите ваши версии библиотечных функций strncpy, strncat и strncmp,

которые обрабатывают не более чем n первых символов своих строковых аргументов.

Например, функция strncpy(s,t,n) копирует не более n символов из строки t в строку s.

Карандашев Н. К.

#include <stdio.h>

#define MAXLEN 1000

int getline(char \*s, int lim);

void strcat(char \*s, char \*t, int n);

void strcpy(char \*s, char \*t, int n);

int strcmp(char \*s, char \*t, int n);

int main()

{

int len, n = 2;

char s[MAXLEN], t[MAXLEN] = "hello";

if ((len = getline(s, MAXLEN)) > 0)

{

strcat(s, t, n);

puts(s);

strcpy(s, t, n);

puts(t);

printf("%d\n", strcmp(s, t, n));

}

return 0;

}

int getline(char \*s, int lim)

{

int c;

char \*ps = s;

while (--lim > 0 && (c = getchar()) != EOF && c != '\n')

\*s++ = c;

if (c == '\n')

\*s++ = c;

\*s = '\0';

return s - ps;

}

void strcat(char \*s, char \*t, int n)

{

while (\*s != '\n')

s++;

while ((\*s++ = \*t++) && (--n != 0))

;

\*s = '\0';

}

void strcpy(char \*s, char \*t, int n)

{

while ((n-- != 0) && \*s)

\*t++ = \*s++;

\*t = '\0';

}

int strcmp(char \*s, char \*t, int n)

{

while ((\*s == \*t) && (\*s != '\0' || \*t != '\0') && (n-- != 0))

s++,t++;

if (\*s == \*t)

return 0;

else if (\*s > \*t)

return 1;

else return -1;

}

**Test input** hi

**Test output** hihe

hi

1

# **Лабораторная работа №12**

**Цель:** Определение характеристик с символьной строки

**Приложение:** 4.3 4.4 4.5 4.6 4.9.c

**К. Р. Упражнение 4.3.** При наличии базовой структуры программы усовершенствование калькулятора уже не представляет особых трудностей.

Реализуйте операцию взятия остатка (%) и работу с отрицательными числами.

**К. Р. Упражнение 4.4.** Добавьте в программу реализацию команд для вывода верхнего элемента стека без его удаления,

для создания в стеке дубликата этого элемента и для обмена местами двух верхних элементов. Также реализуйте команду очистки стека.

**К. Р. Упражнение 4.5.** Добавьте реализацию библиотечных математических функций sin, ехр и pow. См. заголовочный файл <math.h> в приложении Б, раздел 4.

**К. Р. Упражнение 4.6.** Добавьте команды для работы с переменными. (Можно использовать 26 различных переменных, если разрешить имена только из одной буквы.)

Введите переменную, обозначающую последнее выведенное на экран число.

**К. Р. Упражнение 4.9.** Наши функции getch и ungetch не могут корректно обработать символ конца файла EOF, возвращенный в поток ввода.

Подумайте, как эти функции должны реагировать на EOF в буфере, а затем реализуйте ваш замысел.

Юшаков Н. Р. и Карандашев Н. К.

#include <stdio.h> ///заголовочные файлы

#include <stdlib.h>

#include <ctype.h>

#include <math.h>

#include <string.h>

#define MAXOP 100 ///константы

#define NUMBER '0'

#define MAXVAL 100

#define BUFSIZE 100

#define NAME 'n'

int sp = 0; ///глобальные переменные

double val[MAXVAL];

int buf[BUFSIZE];

int bufp = 0;

int getop(char s[]); ///прототипы функций

void push(double);

double pop(void);

int getch(void);

void ungetch(int);

void math(char s[]);

int main ()

{

int type, i, var = 0;

double op2, op1, v, variable[26];

char s[MAXOP];

for (i = 0; i < 26; i++)

variable[i] = 0.0;

while ((type = getop(s)) != EOF)

{

switch(type)

{

case NUMBER:

push(atof(s));

break;

case NAME:

math(s);

case '+':

push(pop() + pop());

break;

case '\*':

push(pop() \* pop());

break;

case '-':

op2 = pop();

push(pop() - op2);

break;

case '/':

op2 = pop();

if (op2 != 0.0)

push(pop() / op2);

else

printf("error: zero divisor\n");

break;

case '%':

op2 = pop();

if (op2 != 0.0)

push((int)pop() % (int)op2);

else

printf("error: zero divisor\n");

break;

case '\_': ///вывод верхнего элемента без его удаления

op2 = pop();

printf("%f\n", op2);

push(op2);

break;

case '=': ///создание дубликата

op2 = pop();

push(op2);

push(op2);

break;

case '$': ///обмен местами 2-х верхних элементов

op1 = pop();

op2 = pop();

push(op1);

push(op2);

break;

case '^': ///очистка стека

sp = 0;

break;

case '@':

pop();

if (var >= 'A' && var <= 'Z')

variable[var - 'A'] = pop();

else

printf("error: invalid name\n");

break;

case '\n':

v = pop();

printf("\t%.8g\n", v);

break;

default:

if (type >= 'A' && type <= 'Z')

push(variable[type - 'A']);

else if (type == 'v')

push(v);

else

printf("error: unknow command %s\n", s);

break;

}

var = type;

}

return 0;

}

void push(double f)

{

if (sp < MAXVAL)

val[sp++] = f;

else

printf("error: stack full, can't push %g\n", f);

}

double pop(void)

{

if (sp > 0)

return val[--sp];

else

{

printf("error: stack empty\n");

return 0.0;

}

}

int getop(char s[])

{

int i = 0, c;

while ((s[0] = c = getch()) == ' ' || c == '\t')

;

s[1] = '\0';

if (islower(c))

{

while (islower(s[++i] = c = getch()))

;

s[i] = '\0';

if (c != EOF)

ungetch(c);

if (strlen(s) > 1)

return NAME;

else

return c;

}

if (!isdigit(c) && c != '.' && c != '-')

return c;

if (c == '-')

if (isdigit(c = getch()) || c == '.')

s[++i] = c;

else

{

if (c != EOF)

ungetch(c);

return '-';

}

if (isdigit(c))

while (isdigit(s[++i] = c = getch()))

;

if (c == '.')

while (isdigit(s[++i] = c = getch()))

;

s[i] = '\0';

if (c != EOF)

ungetch(c);

return NUMBER;

}

int getch(void)

{

return (bufp > 0) ? buf[--bufp] : getchar();

}

void ungetch(int c)

{

if (bufp >= BUFSIZE)

printf("ungetch: too many characters\n");

else

buf[bufp++] = c;

}

void math(char s[])

{

double op2;

if (strcmp(s, "sin") == 0)

push(sin(pop()));

else if (strcmp(s, "cos") == 0)

push(cos(pop()));

else if (strcmp(s, "exp") == 0)

push(exp(pop()));

else if (strcmp(s, "pow") == 0)

{

op2 = pop();

push(pow(pop(), op2));

}

else

printf("error: %s invalid command\n", s);

}

**Test input** 2 2 + 2 \*

**Test output** 8

# **Лабораторная работа №13**

# **Лабораторная работа №14**

**Цель:** Изучение работы системы Linux

Приложение: Hello.FODT, Stepik.org курс «Введение в Linux»

**Раздел 1.3 задание 1:** Создайте документ в OpenOffice/LibreOffice Writer (аналог Microsoft Word) и напишите в нём шрифтом FreeMono (если такого шрифта у вас нет, то используйте Arial или Times New Roman) одну-единственную строчку:

1. Hello, Linux!

После этого сохраните этот документ в формате XML (Microsoft Word 2003 XML) или в формате FODT (OpenDocument Text: Flat XML)

**Раздел 1.3 задание 2 (тест):** Какое расширение имеют установочные пакеты в Linux (Ubuntu)?

1. deb

**Раздел 1.3 задание 3:** Поставьте себе в систему плеер VLC (любым способом: через Software Center или скачиванием установочного пакета с сайта VLC).

Запустите, откройте Help → About (или Shift+F1) и напишите ниже первую фамилию (без имени!) из вкладки Authors. Обратите внимание, что в англоязычных текстах обычно имя стоит на первом месте (first name), а фамилия на втором (last name).

1. Denis-Courmont

**Раздел 1.3 задание 4 (тест):** Для чего можно использовать приложение Update Manager?

1. Для обновления установленных программ
2. Для обновления всей системы до новой версии
3. Для обновления ссылок в Software Center

# **Лабораторная работа №15**

**Цель:** Изучение работы в консоли

**Приложение:** Stepik.org курс «Введение в Linux»

**Раздел 1.4 задание 4 (тест):** Предположим, что вы находитесь в директории */home/bi/Documents*, причем */home/bi* — ваша домашняя директория. Какая(ие) команда выведет содержимое */home/bi/Downloads*, при этом не показывая содержимое других директорий?

1. ls /home/bi/Downloads

**Раздел 1.5 задание 1 (тест):** Что произойдет, если ввести в терминал команду firefox (для запуска одноименного браузера), а затем ввести туда же команду exit?

1. Никто не закроется

**Раздел 1.6 задание 2 (тест):** Какие (какая) из команд создадут файл file.txt и запишут в него поток ошибок программы program? Считайте, что в момент запуска программы файл file.txt не существует.

1. program 2>> file.txt
2. program 2> file.txt

**Раздел 1.7 задание 1 (тест):** В каком файле на диске окажется картинка, если для её скачивания были выполнены следующие команды?

1. cd /home/alex/
2. wget -P /home/alex/Pictures -O 1.jpg http://example.com/example.jpg
3. /home/alex/1.jpg

**Раздел 1.8 задание 3 (тест):** Какой набор опций нужно указать программе tar, чтобы запаковать файлы в my\_archive.tar.bz2?

1. -cjf

**Раздел 1.9 задание 2 (тест):** Предположим, что в файле text.txt записаны строки, показанные среди вариантов ответа. Отметьте только те из них, которые выведет на экран команда grep "world" text.txt.

1. The "world" is not enough
2. The beautifulworld is not enough
3. The world is not enough
4. The beautiful-world is not enough
5. world

**Раздел 2.2 задание 2 (тест):** Предположим, что вы устанавливаете программу program на свой компьютер при помощи команды sudo apt-get install program. Терминал сообщает вам, что он не может найти и скачать установочный пакет. Какие действия могут устранить проблему?

1. Проверка интернет соединения и его установка, если соединения нет.
2. sudo apt-get update

# **Лабораторная работа №16**

**Цель:** Работа в редакторе vim

**Приложение:** Stepik.org курс «Введение в Linux»

**Раздел 3.1 задание 4:** Предположим, что вы открыли файл в редакторе vim и хотите заменить в этом файле все строки, содержащие слово Windows, на такие же строки, но со словом Linux. Если в какой-то строке слово Windows встречается больше, чем один раз, то заменить на Linux в этой строке нужно только самое первое из этих слов. Какую команду нужно ввести для этого в vim? Укажите необходимую команду целиком (т.е. включая ввод ":" в самом начале), однако нажатие на Enter после ввода команды обозначать никак не нужно.

*:%s/Windows/Linux*

**Раздел 3.1 задание 1(тест):** Какую клавишу(и) нужно нажать на клавиатуре, чтобы выйти из редактора vim? Считайте, что вы только что открыли файл и вам сразу понадобилось выйти из редактора.

1. ":", затем "q", затем "Еnter"

# **Лабораторная работа №17**

# **Лабораторная работа №18**

**Цель:** Создание динамической библиотеки

**Приложение:** Stepik.org курс «Основы программирования на Linux» solution.c Makefile

**Раздел 1.1 задание 3:** В этом задании Вам необходимо создать разделяемую динамическую библиотеку libsolution.so в которой реализовать функцию со следующим прототипом:

int stringStat(const char \*string, size\_t multiplier, int \*count);

Функция возвращает значение длины строки string, умноженное на multiplier, и увеличивает на 1 значение, на которое указывает count.

Решение предоставляется в виде двух файлов solution.c и Makefile, в последнем предполагается цель по умолчанию, которая приводит к сборке Вашей библиотеки.

Solution.c

#include <stdio.h>

#include <stddef.h>

#include <string.h>

int stringStat(const char \*string, size\_t multiplier, int \*count)

{

int i = 0;

i = strlen(string);

(\*count)++;

return i \* multiplier;

}

Makefile

libsolution.so: solution.c

gcc -o libsolution.so -shared -fPIC solution.c

**Раздел 1.2 задание 1(тест):** Проверьте свои знания. Отметьте верные высказывания.

1. Для вывода символических имен объектного(бинарного) файла используется nm
2. Для сборки динамической библиотеки gcc требуется указать опцию –shared
3. В основе работы make лежит информация о зависимостях частей проекта
4. с++filt используется для восстановления исходного имени функций (Demangle) на языках на C++ и Java

# **Лабораторная работа №19**

**Цель:** Работа с Libdl

**Приложение:** Stepik.org курс «Основы программирования на Linux»

**Раздел 1.3 задание 1:** Вам требуется скомпилировать программу, состоящую из единственного файла main.c в исполняемый файл loader. Вы знаете, что данная программа использует libdl для загрузки динамических библиотек в run-time. Какая команда, на ваш взгляд, для этого больше всего подходит?

gcc main.c -fPIC -ldl -o loader

**Раздел 1.3 задание 2:** Проверьте свои знания, отметьте верные утверждения.

1. Программист должен вручную привести указатель возвращаемый dlsym к типу вызываемой функции

# **Лабораторная работа №20**

# **Лабораторная работа №21**

**Цель:** Работа с файлами и каталогами

**Приложение:** Stepik.org курс «Основы программирования на Linux»

**Раздел 2.1 задание 1(тест):** Отметьте функции, которые на Ваш взгляд, относятся к системному интерфейсу доступа к файлам в Linux

1. int open(const char \*pathname, int flags);
2. int creat(const char \*pathname, mode\_t mode)
3. ssize\_t write(int fd, const void \*buf, size\_t count)
4. int open(const char \*pathname, int flags, mode\_t mode)

**Раздел 2.2 задание 1(тест):** Вы пишете программу для обхода всех каталогов вашей файловой системы и подсчета количества файлов. Какое условие наиболее подходит для того чтобы прекратить обход?

1. readdir вернул NULL

# **Лабораторная работа №22**

# **Лабораторная работа №23**

**Цель:** Работа с процессами

**Приложение:** Stepik.org курс «Основы программирования на Linux» 3.5.1.с 3.5.2.с 3.5.3.с

**Раздел 3.5 задание 1:** Разработать программу solution, которая осуществляет поиск родительского PID текущего процесса в файловой системе proc и выводит найденное значение на консоль. Результат, возвращаемый функций должен быть выведен отдельной строкой(должен оканчиваться символом перевода строки \n) в stdio.

#include <sched.h>

#include <sys/types.h>

#include <stdio.h>

#include <string.h>

int main()

{

int ppid;

char tp[255] = { 0 };

FILE\* f = fopen("/proc/self/stat", "r");

fscanf(f, "%\*[^)] %\*c %\*c %d %\*s", &ppid);

printf("%d\n", ppid);

return 0;

}

**Раздел 3.5 задание 2:** Разработать программу solution, которая осуществляет поиск и подсчет числа процессов с именем genenv в системе Linux на которой выполняется программа.

#include <stdio.h>

#include <unistd.h>

int main()

{

system("pidof -c genenv | wc -w");

}

**Раздел 3.5 задание 3:** Разработать программу solution, которая по заданному pid, осуществляет поиск пути в дереве процессов до процесса с идентификатором 1 (init). Для каждого найденного процесса печатается в отдельной строке его идентификатор.

#include <sys/types.h>

#include <unistd.h>

#include <stdio.h>

#include <stdlib.h>

int main(int argc, char \*\*argv)

{

int pid = atoi(argv[1]);

char tmp[255] = { 0 };

printf("%d\n", pid);

while (pid != 1)

{

sprintf(tmp, "/proc/%d/stat", pid);

FILE\* f = fopen(tmp, "r");

fscanf(f, "%\*[^)] %\*c %\*c %d %\*s", &pid);

printf("%d\n", pid);

fclose(f);

}

return 0;

}

# **Лабораторная работа №24**

**Цель:** Работа с процессами

**Приложение:** Stepik.org курс «Основы программирования на Linux» 3.5.1.с 3.5.2.с 3.5.3.с

**Раздел 3.5 задание 1:** Разработать программу solution, которая осуществляет поиск родительского PID текущего процесса в файловой системе proc и выводит найденное значение на консоль. Результат, возвращаемый функций должен быть выведен отдельной строкой(должен оканчиваться символом перевода строки \n) в stdio.

#include <sched.h>

#include <sys/types.h>

#include <stdio.h>

#include <string.h>

int main()

{

int ppid;

char tp[255] = { 0 };

FILE\* f = fopen("/proc/self/stat", "r");

fscanf(f, "%\*[^)] %\*c %\*c %d %\*s", &ppid);

printf("%d\n", ppid);

return 0;

}

**Раздел 3.5 задание 2:** Разработать программу solution, которая осуществляет поиск и подсчет числа процессов с именем genenv в системе Linux на которой выполняется программа.

#include <stdio.h>

#include <unistd.h>

int main()

{

system("pidof -c genenv | wc -w");

}

**Раздел 3.5 задание 3:** Разработать программу solution, которая по заданному pid, осуществляет поиск пути в дереве процессов до процесса с идентификатором 1 (init). Для каждого найденного процесса печатается в отдельной строке его идентификатор.

#include <sys/types.h>

#include <unistd.h>

#include <stdio.h>

#include <stdlib.h>

int main(int argc, char \*\*argv)

{

int pid = atoi(argv[1]);

char tmp[255] = { 0 };

printf("%d\n", pid);

while (pid != 1)

{

sprintf(tmp, "/proc/%d/stat", pid);

FILE\* f = fopen(tmp, "r");

fscanf(f, "%\*[^)] %\*c %\*c %d %\*s", &pid);

printf("%d\n", pid);

fclose(f);

}

return 0;

}

# **Лабораторная работа №25**

# **Лабораторная работа №26**

**Цель:** Работа с каналами

**Приложение:** Stepik.org курс «Основы программирования на Linux» 4.6.2.с

**Раздел 4.6 задание 2:** В текущем каталоге есть 2 канала in1 in2, в которые в случайном порядке поступают числа, которые необходимо просуммировать и вывести окончательную сумму на экран. Сумма выводится в отдельной строке, завершающейся символом конца строки. Признаком окончания подачи символов в каналы является закрытие этих каналов посылающей стороной.

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <sys/select.h>

#include <stddef.h>

#include <unistd.h>

#include <stdio.h>

#include <stdlib.h>

void calculate\_input(int fd, int \*sum, fd\_set \*list) {

char buf[128];

int bytes = read(fd, buf, sizeof(buf) - 1);

if (bytes == 0) {

close(fd);

FD\_CLR(fd, list);

return;

}

buf[bytes] = 0;

\*sum += atoi(buf);

}

int main(int argc, char const \*argv[])

{

int sum = 0;

int fd1 = open("./in1", O\_RDONLY|O\_NONBLOCK);

int fd2 = open("./in2", O\_RDONLY|O\_NONBLOCK);

if (! (fd1 && fd2)) {

printf("Fifo cannot open");

return 1;

}

fd\_set wathc\_list;

fd\_set in\_list;

FD\_ZERO(&wathc\_list);

FD\_SET(fd1, &wathc\_list);

FD\_SET(fd2, &wathc\_list);

while(FD\_ISSET(fd1, &wathc\_list) || FD\_ISSET(fd2, &wathc\_list)) {

in\_list = wathc\_list;

int res = select(fd2+1, &in\_list, NULL, NULL, NULL);

if (res) {

if(FD\_ISSET(fd2, &in\_list)) {

calculate\_input(fd2, &sum, &wathc\_list);

}

if(FD\_ISSET(fd1, &in\_list)) {

calculate\_input(fd1, &sum, &wathc\_list);

}

}

}

printf("%d", sum);

return 0;

}

# **Лабораторная работа №27**

**Цель:** Работа с сигналами

**Приложение:** Stepik.org курс «Основы программирования на Linux» 4.5.3.с

**Раздел 4.6 задание 3:** Разработать приложение, умеющее обрабатывать сигналы SIGUSR1 ,SIGUSR2, SIGTERM. После старта Ваше приложение должно по приходу одного из сигналов SIGUSR1, SIGUSR2 выполнять суммирование числа срабатываний каждого из сигналов, а после прихода сигнала SIGTERM, требуется вывести в стандартный поток вывода 2 числа, разделенных пробелом, соответствующих количеству обработанных сигналов SIGUSR1, SIGUSR2, и завершить программу. Вывод оканчивается символом конца строки.

#include <stdio.h>

#include <stdlib.h>

#include <signal.h>

#include <sched.h>

#include <unistd.h>

static int cnt1 = 0;

static int cnt2 = 0;

static int cond = 1;

void h1()

{

++cnt1;

}

void h2()

{

++cnt2;

}

void h3()

{

cond = 0;

}

int main()

{

signal(SIGUSR1,h1);

signal(SIGUSR2,h2);

signal(SIGTERM,h3);

while(cond)

sleep(100);

printf("%d %d\n", cnt1, cnt2);

return 0;

}

# **Лабораторная работа №28**