Национальный исследовательский ядерный университет «МИФИ»

Институт интеллектуальных кибернетических систем

Кафедра №12 «Компьютерные системы и технологии»

|  |  |  |
| --- | --- | --- |
| Изображение выглядит как графическая вставка, Графика, графический дизайн, мультфильм  Автоматически созданное описание | ИИКС НИЯУ МИФИ — Институт интеллектуальных кибернетических систем | Изображение выглядит как снимок экрана, Графика, дизайн  Автоматически созданное описание |

**ОТЧЕТ**

**О выполнении лабораторной работы №4**

**«Работа со строками»**

Не хватает: таблица с тестами и скриношоты работы проги

**Студент:** Рыженко Р.В.

**Группа:** Б23-506

**Преподаватель:** Курочкина М-А.А.

*Москва 2023*

1. **Формулировка индивидуального задания**

Вариант №5.

Осуществить циклический сдвиг на N символов влево в каждом слове строки.

1. **Описание использованных типов данных**

При выполнении данной лабораторной работы использовался встроенные типы данных double и int, предназначенные для работы с вещественными и целыми числами, а также char для работы с символами и строками и указатели, предназначенные для работы с адресами в памяти.

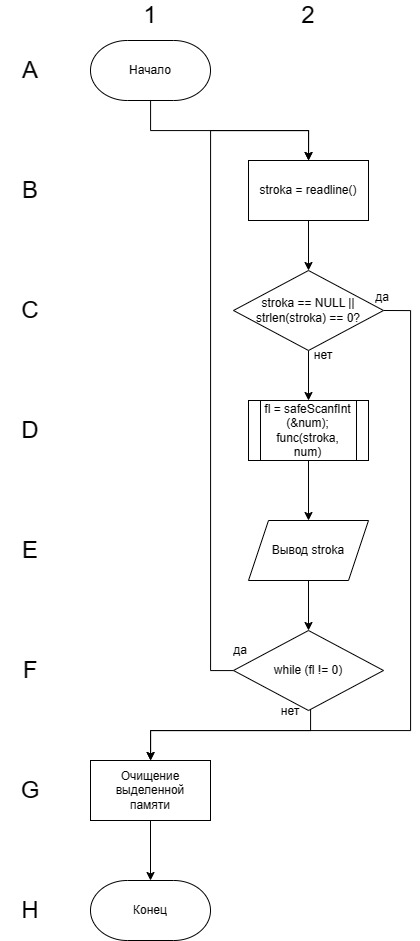
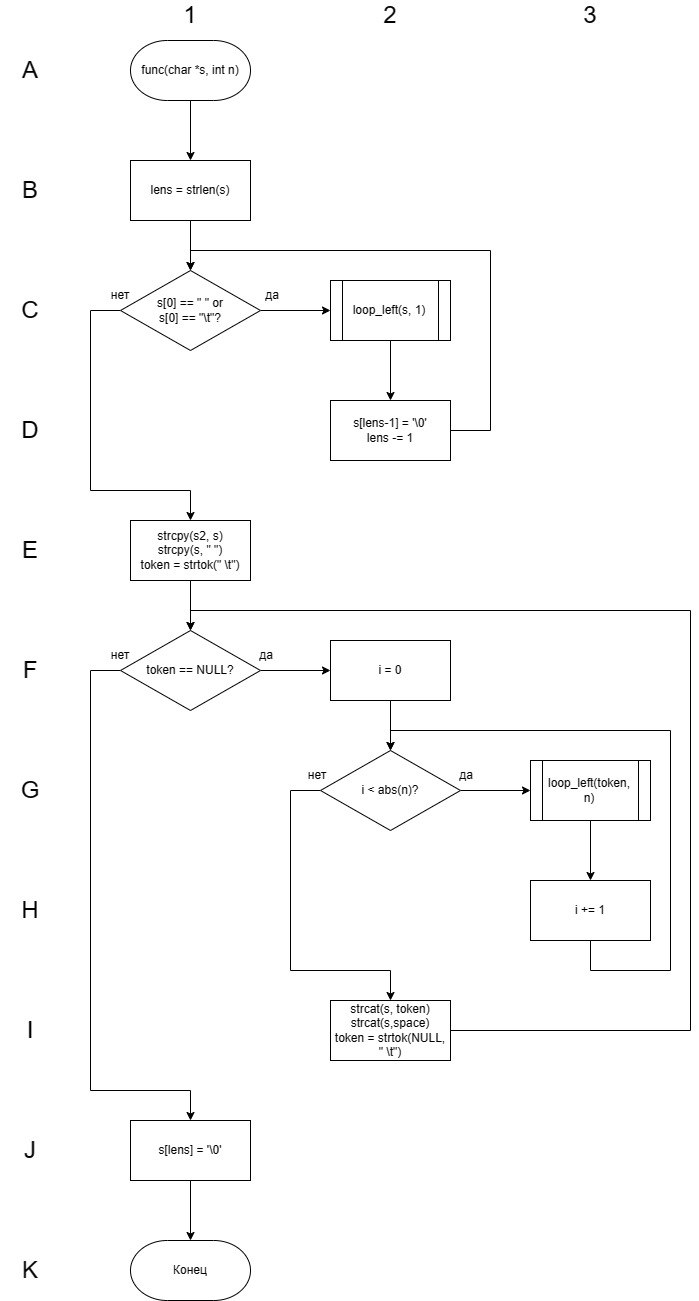
1. **Описание использованного алгоритма**

Рис. 1: Блок-схема алгоритма работы функции main()



­­­

Рис. 2: Блок-схема алгоритма работы функции func()

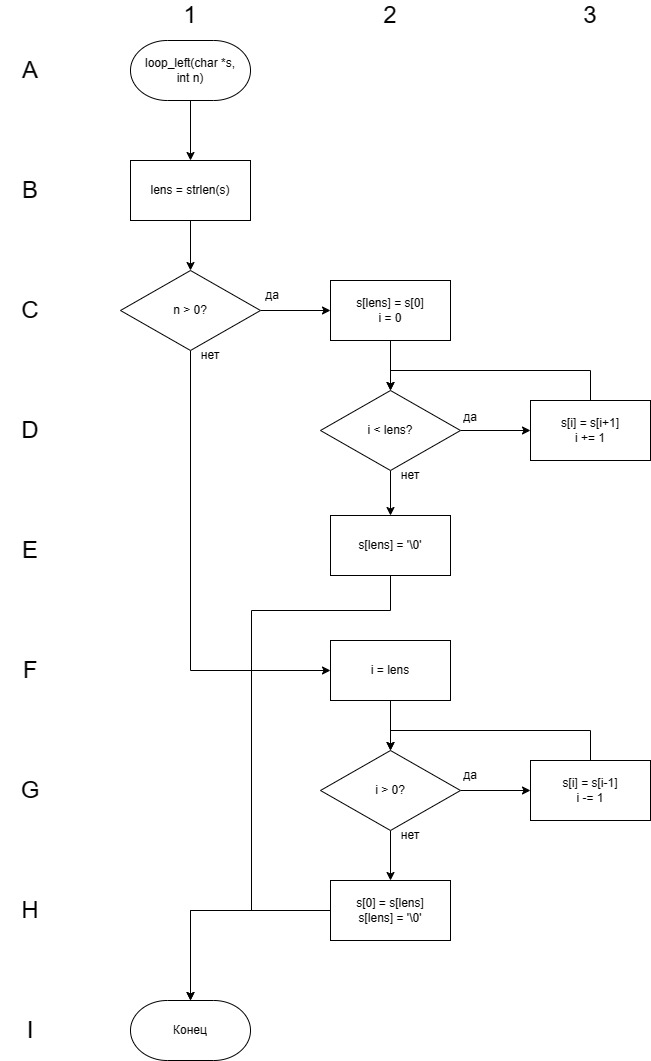


Рис. 3: Блок-схема алгоритма работы функции loop\_left()

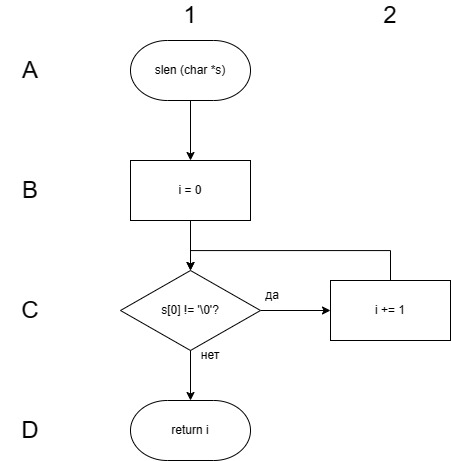


Рис. 4: Блок-схема алгоритма работы функции slen()

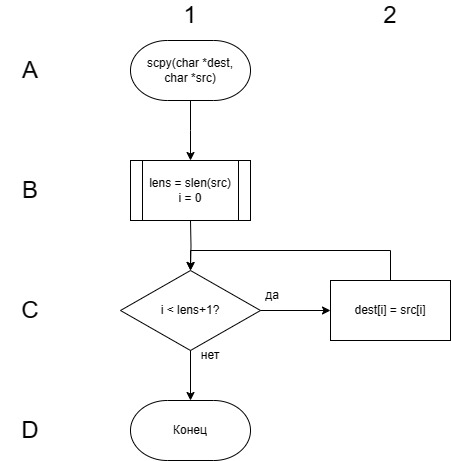


Рис. 5: Блок-схема алгоритма работы функции scpy()

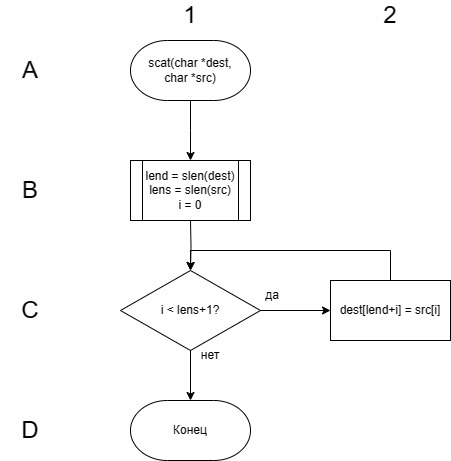


Рис. 6: Блок-схема алгоритма работы функции scat()

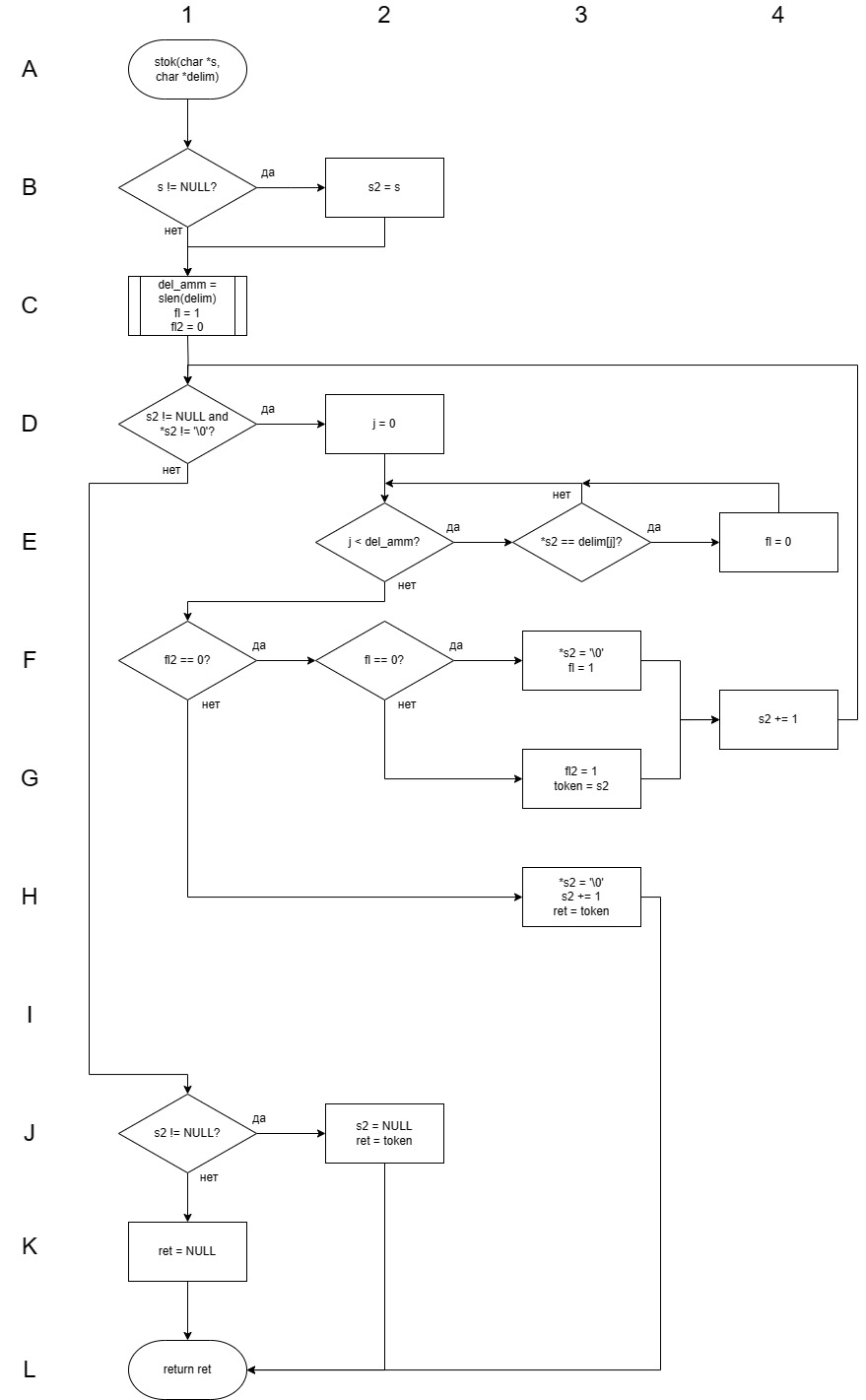


Рис. 7: Блок-схема алгоритма работы функции stok()

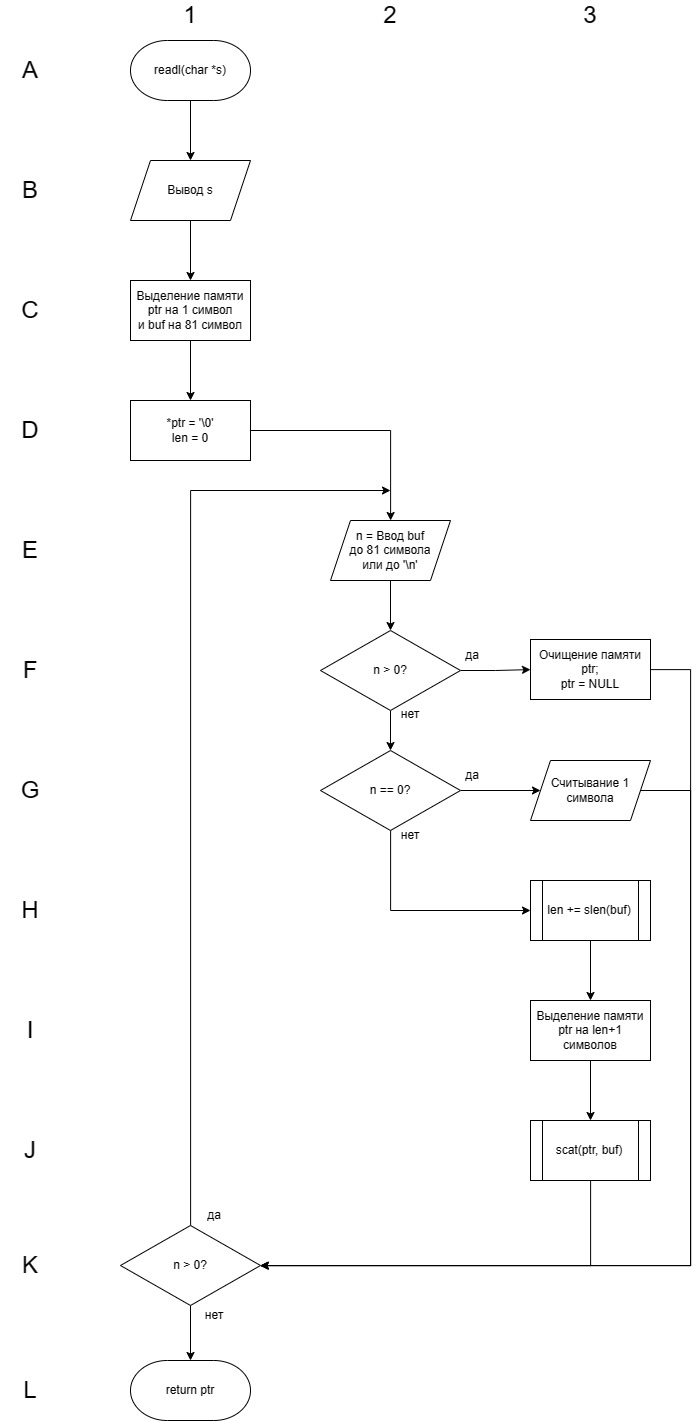


Рис. 8: Блок-схема алгоритма работы функции readl()

1. **Исходные коды разработанных программ**

Листинг 1: Исходный код функции main() (файл: main.c)

#include <stdio.h>

#include <readline/readline.h>

#include <stdlib.h>

#include <string.h>

#include "myfuncs.h"

int main()

{

char \*stroka = NULL;

int number;

int fl;

do {

stroka = readline("Input a string\n");

printf("\n");

if (stroka == NULL || strlen(stroka) == 0) {

break;

}

printf("Input a number\n");

fl = safeScanfInt(&number);

printf("\n");

if (fl == 0) {

break;

}

func(stroka, number);

printf("Output string:\n");

printf("\"%s\"\n\n", stroka);

} while (stroka != NULL && fl != 0);

printf("Ending the program\n");

free(stroka);

return 0;

}

Листинг 2: Исходный код функций для индивидуального задания (файл myfuncs.c)

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "myfuncs.h"

#define space " "

#define tab "\t"

#define spacetab " \t"

int safeScanfInt (int \*target)

{

int guard;

int flag = 1;

while (flag == 1) {

guard = scanf("%d",target);

scanf("%\*[^\n]");

scanf("%\*c");

if (guard == EOF) {

return 0;

}

if (guard < 1) {

printf("Please input an integer number!\n");

continue;

}

flag = 0;

}

return 1;

}

void loop\_left(char \*s, int n)

{

int lens = strlen(s);

int i;

if (n > 0) { // сдвиг влево

s[lens] = s[0];

for (i = 0; i < lens; i++) {

s[i] = s[i+1];

}

s[lens] = '\0';

}

else { // сдвиг вправо

for (i = lens; i > 0; i--) {

s[i] = s[i-1];

}

s[0] = s[lens];

s[lens] = '\0';

}

}

void func(char \*s, int n)

{

int lens = strlen(s);

while (s[0] == space[0] || s[0] == tab[0]) {

loop\_left(s, 1);

s[lens-1] = '\0';

lens--;

}

char s2[lens];

strcpy(s2, s);

strcpy(s, "");

char \*token;

token = strtok(s2, spacetab);

int i;

while (token != NULL) {

for (i = 0; i < abs(n); i++) {

loop\_left(token, n);

}

strcat(s, token);

token = strtok(NULL, spacetab);

if (token != NULL {

strcat(s, space);

}

}

// free(s2);

}

Листинг 3: Исходный код файла myfuncs.h

#ifndef MYFUNCS\_H

#define MYFUNCS\_H

int safeScanfInt (int\*);

void loop\_left(char\*, int);

void func(char\*, int);

#endif

Листинг 4: Исходный код сделанных мною функций библиотеки string.h (файл: mystringh.c)

#include <stdio.h>

#include <stdlib.h>

#include "mystringh.h"

int slen (char \*s)

{

int i=0;

while (s[i] != '\0') {i++;}

return i;

}

void scpy (char \*dest, char \*src) // предполагается, что длина dest >= длина src

{

int lens = slen(src);

int i;

for (i = 0; i < lens+1; i++) {

dest[i] = src[i];

}

}

void scat (char \*dest, char \*src)

{

int lend = slen(dest);

int lens = slen(src);

int i;

for (i = 0; i < lens+1; i++) {

dest[lend + i] = src[i];

}

}

char \*readl(char \*s)

{

printf(s);

char \*ptr = (char\*)malloc(1);

\*ptr = '\0';

char buf[81];

int n, len=0;

do {

n = scanf("%80[^\n]", buf);

if (n < 0) {

free(ptr);

ptr = NULL;

continue;

}

if (n == 0) {

scanf("%\*c");

}

else {

len += slen(buf);

ptr = (char\*)realloc(ptr, len+1);

scat(ptr, buf);

}

} while (n > 0);

return ptr;

}

char \*stok (char \*s, char \*delim)

{

static char \*s2 = NULL;

char \*token = NULL;

if (s != NULL) {

s2 = s;

}

int del\_amm = slen(delim);

int fl, fl2;

int j;

fl = 1; fl2 = 0; // fl = "s[i] - буква", fl2 = "ранее уже нашли букву"

while (s2 != NULL && \*s2 != '\0') {

for (j = 0; j < del\_amm; j++) {

if (\*s2 == delim[j]) {

fl = 0;

}

}

if (fl2 == 0) { // ещё не нашли букву ранее

if (fl == 0) { // s[i] - пробел

\*s2 = '\0';

fl = 1; // ресет флага перед новым циклом

}

else { // fl == 1; s[i] - буква

fl2 = 1; // нашли первую букву

token = s2;

}

}

else /\*fl2 == 1\*/ if (fl == 0) { // после нахождения первой буквы снова нашли пробел

\*s2 = '\0';

s2++;

return token;

}

s2++;

}

if (s2 != NULL) {

s2 = NULL;

return token;

}

return NULL; // if s2 == NULL

}

Листинг 5: Исходный код файла mystringh.h

#ifndef MYSTRINGH\_H

#define MYSTRINGH\_H

int slen (char\*);

void scpy (char\*, char\*);

void scat (char\*, char\*);

char \*readl(char\*);

char \*stok (char\*, char\*);

#endif

1. **Описание тестовых примеров**

Таблица 1: Тестовые примеры

|  |  |  |
| --- | --- | --- |
| Ввод строки | Ввод числа | Вывод строки |
| “abc def” | 0 | “abc def” |
| “abc def” | 1 | “bcd efd” |
| “abc def” | -1 | “cab fde” |
| “abc def” | 3 | “abc def” |
| “abc def” | 4 | “bcd efd” |
| “ abcdefg hi jkl “ | 2 | “cdefgab hi ljk” |
| Ctrl+D |  | Ending the program |

1. **Скриншоты**

Рис. 9: Сборка программы pr1

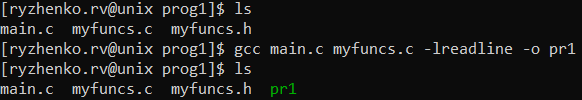


Рис. 10: Сброка программы pr2

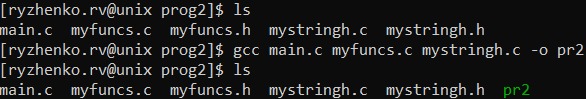


Рис. 11: Запуск программы pr1

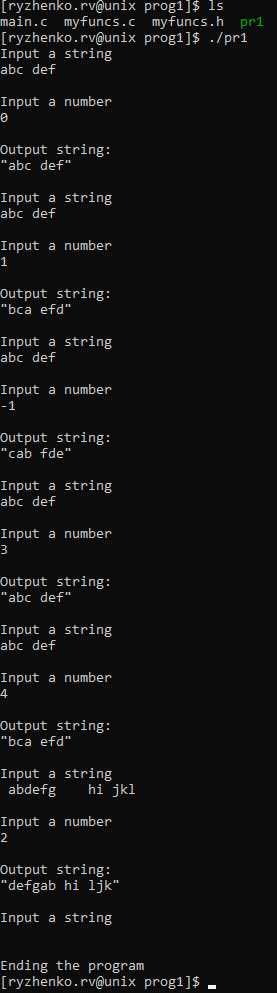
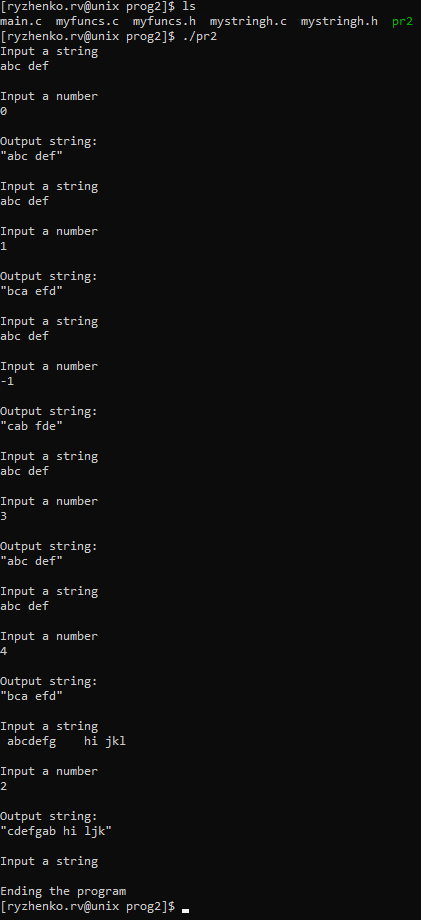


Рис. 12: Запуск программы pr2



1. **Выводы**

В ходе выполнения данной работы на примере программы, выполняющей работу со строками, были рассмотрены базовые принципы работы построения программ на языке C, обработки целых и вещественных чисел, символов и строк:

1. Организация ввода/вывода.
2. Разработка функций.
3. Объявление и использование переменных.
4. Выполнение простейших арифметических операций над целочисленными и вещественными операндами.
5. Операции над символами и строками.