

Национальный исследовательский ядерный университет «МИФИ»

Институт интеллектуальных кибернетических систем

Кафедра №12 «Компьютерные системы и технологии»



ОТЧЕТ

**О выполнении лабораторной работы №4
«Работа со строками»**

Студент: Мартыненко Д. М.

Группа: С22-501

Преподаватель: Овчаренко Е. С.

Москва — 2022

1. Формулировка индивидуального задания

Вариант 15: осуществить циклический сдвиг на N символов влево в каждом слове строки.

2. Описание использованных типов данных

При выполнении лабораторной работы были использованны следующие типы данных:

1. Тип `int` для `for-loop` циклов, а также для других подсчётов.
2. Тип `char` для отдельных символов строки, то есть элементов массива `char*`.

3. Описание использованного алгоритма

3.1. С использованием библиотечных функций

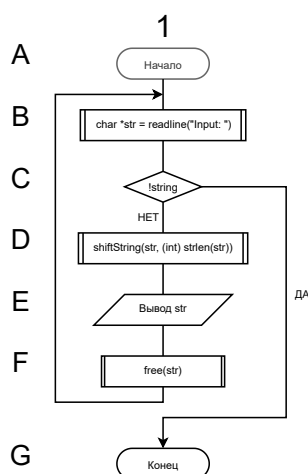


Рис. 1: Блок-схема алгоритма работы функции `main()` в библиотеке `default`

3.2. Без использования библиотечных функций

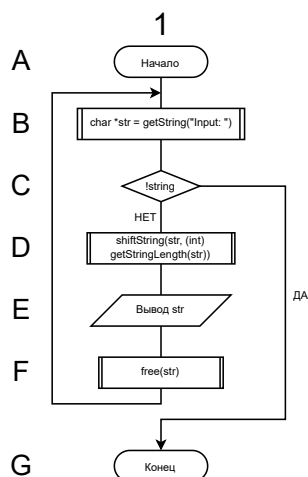


Рис. 2: Блок-схема алгоритма работы функции `main()` в библиотеке `custom`

3.2.1. Аналоги функций из библиотеки `readline.h`

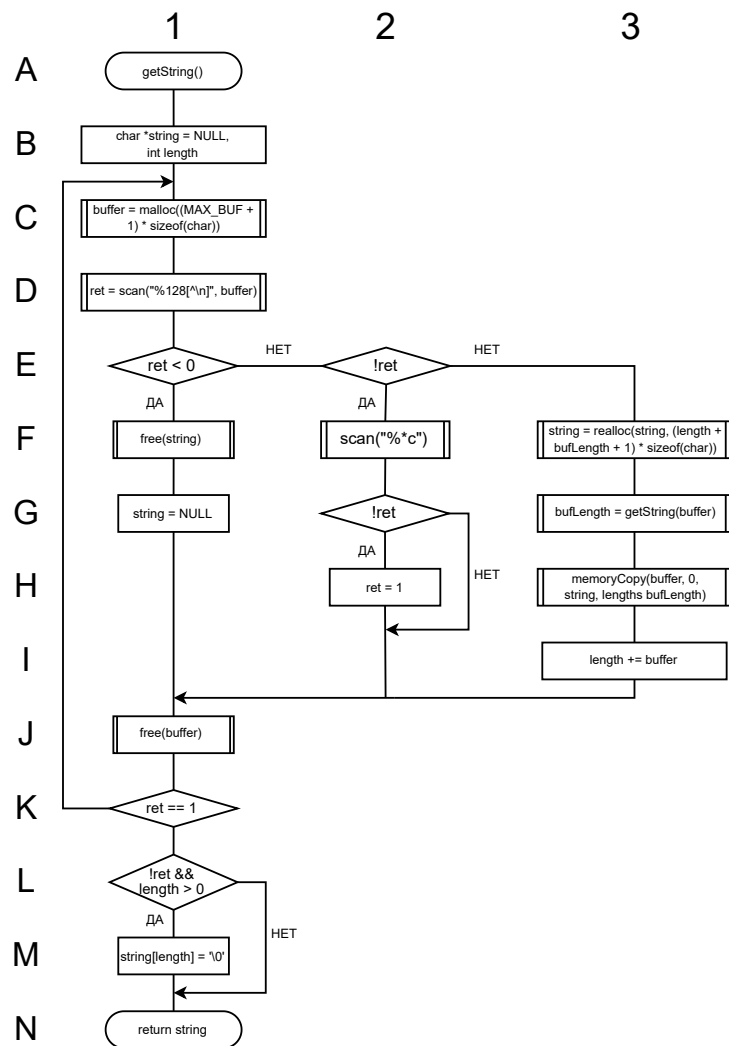


Рис. 3: Блок-схема алгоритма работы функции `getString()`

3.2.2. Аналоги функций из библиотеки `string.h`

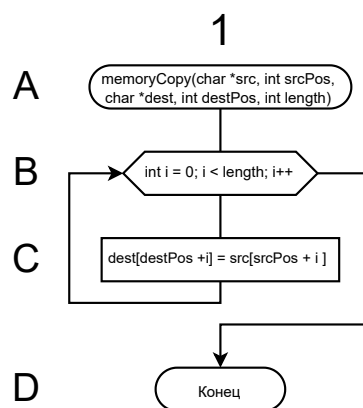


Рис. 4: Блок-схема алгоритма работы функции `memoryCopy()`

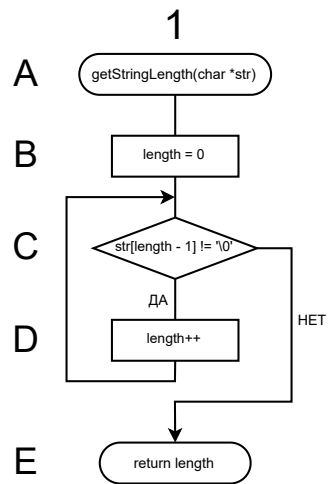


Рис. 5: Блок-схема алгоритма работы функции `getStringLength()`

3.3. Описание алгоритма модификации исходной строки

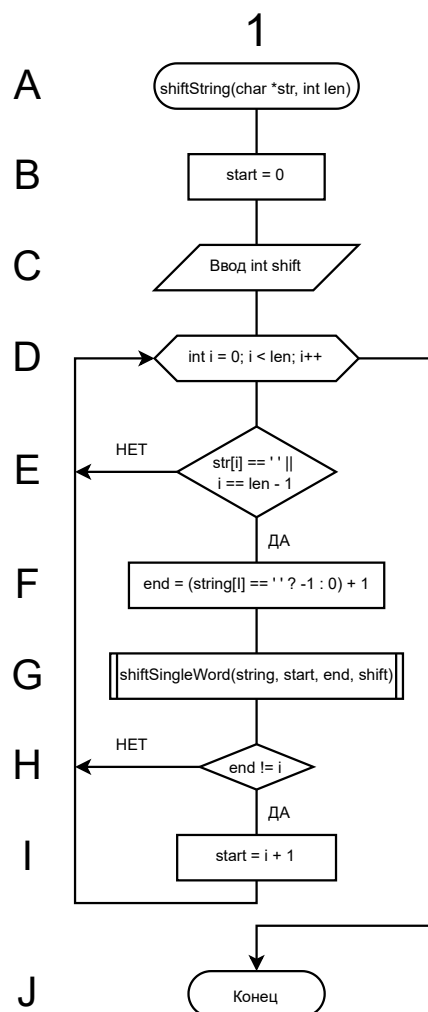


Рис. 6: Блок-схема алгоритма работы функции `shiftString()`

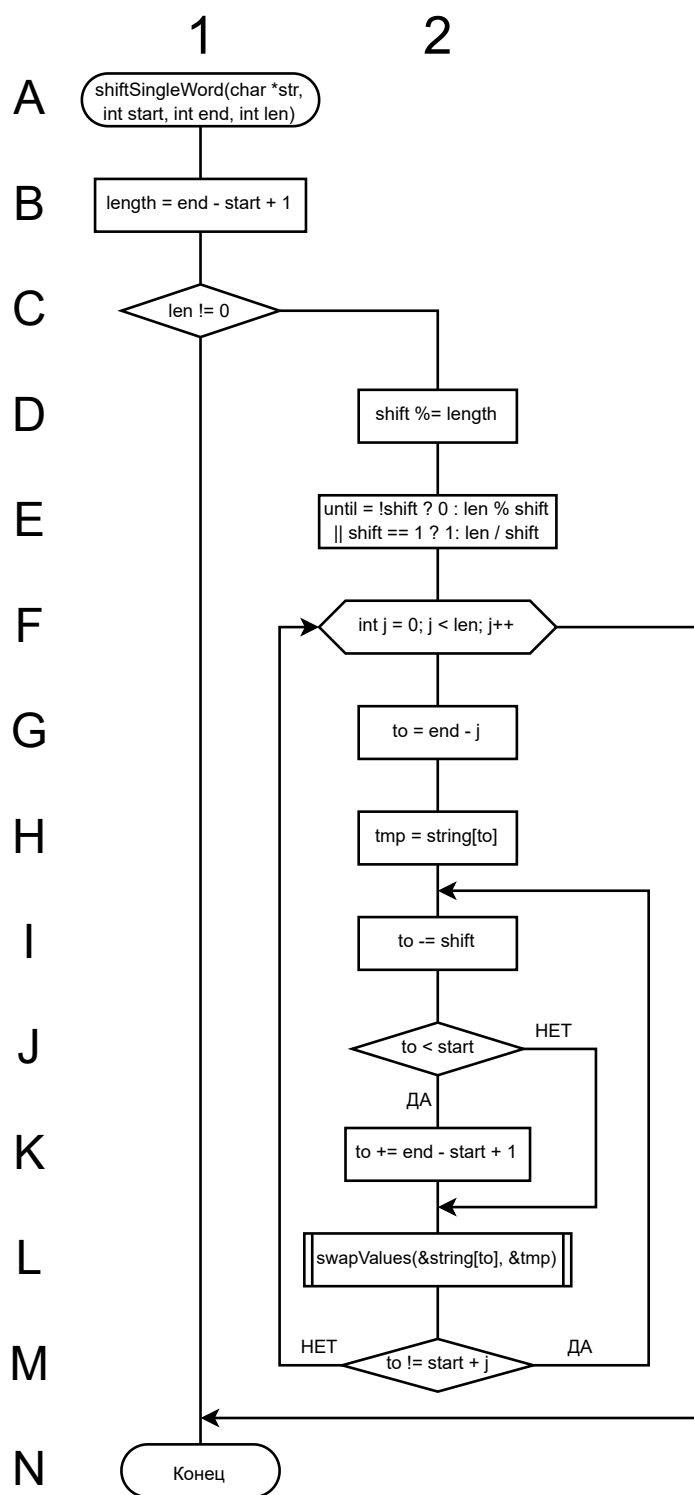


Рис. 7: Блок-схема алгоритма работы функции shiftSingleWord()

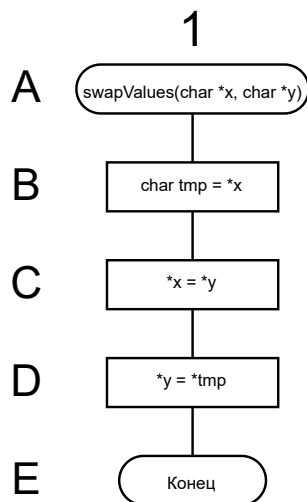


Рис. 8: Блок-схема алгоритма работы функции swapValues ()

4. Исходные коды разработанных программ

4.1. Исходный код программы с использованием библиотек

Исходный код 1: библиотека default.c

```

1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4  #include <time.h>
5  #include <readline/readline.h>
6
7  #include "include/output.h"
8  #include "include/parse.h"
9
10 int main() {
11     char *string;
12
13     clearOutput();
14     while (1) {
15         clock_t time = clock();
16         string = readline(INPUT_MSG);
17         if (!string) break;
18         shiftString(string, (int) strlen(string));
19         time = clock() - time;
20         printf("Result: %s\n", string);
21         printf("Time perform shift: %Lf\n", (long double) time / CLOCKS_PER_SEC);
22         free(string);
23         printDelimiterMsg();
24     }
25     printExitMsg();
26
27     return 0;
28 }
  
```

4.2. Исходный код программы без использованием библиотек

Исходный код 2: библиотека custom.c

```

1  #include <stdio.h>
  
```

```

2 #include <stdlib.h>
3 #include <time.h>
4
5 #include "include/output.h"
6 #include "include/input.h"
7 #include "include/parse.h"
8 #include "include/utils.h"
9
10 int main() {
11     char *string;
12
13     clearOutput();
14     while (1) {
15         clock_t time = clock();
16         printf(INPUT_MSG);
17         string = getString();
18         if (!string) break;
19         shiftString(string, getStringLength(string));
20         time = clock() - time;
21         printf("Result: %s\n", string);
22         printf("Time perform shift: %Lf\n", (long double) time / CLOCKS_PER_SEC);
23         free(string);
24         printDelimiterMsg();
25     }
26     printExitMsg();
27
28     return 0;
29 }

```

4.3. Исходный код модификации строки

Исходный код 3: библиотека parse.c.

```

1 #include <stdio.h>
2 #include <time.h>
3
4 #include "include/parse.h"
5 #include "include/output.h"
6
7 void shiftString(char *string, int length) {
8     int shift, start = 0;
9
10    printf("Enter required shift value: ");
11    while (!scanf("%d", &shift)) {
12        printErrorMsg("This is invalid input, try again: ");
13        scanf("%*[^\\n]");
14    }
15    scanf("%*c");
16
17    for (int i = 0; i < length; i++) {
18        if (string[i] == ' ' || i == length - 1) {
19            int end = (string[i] == ' ' ? -1 : 0) + i;
20            shiftSingleWord(string, start, end, shift);
21            if (end != i) start = i + 1;
22        }
23    }
24 }
25
26 static void shiftSingleWord(char *string, int start, int end, int shift) {
27     int length = end - start + 1;
28     if (length != 0) {

```

```

29     shift = shift * (shift < 0 ? -1 : 1) % length;
30     int until = !shift ? 0 : length % shift || shift == 1 ? 1 : length / shift;
31     for (int j = 0; j < until; j++) {
32         int to = end - j;
33         char tmp = string[to];
34         do {
35             to -= shift;
36             if (to < start) to += end - start + 1;
37             swapValues(&string[to], &tmp);
38         } while (to != end - j);
39     }
40 }
41 }
42
43 static void swapValues(char *x, char *y) {
44     char tmp = *x;
45     *x = *y;
46     *y = tmp;
47 }

```

Исходный код 4: заголовочный файл include/parse.h.

```

1 #ifndef INCLUDE_PARSE_H
2 #define INCLUDE_PARSE_H
3
4 void shiftString(char *, int);
5
6 static void shiftSingleWord(char *, int, int, int);
7 static void swapValues(char *, char *);
8
9 #endif //INCLUDE_PARSE_H

```

4.4. Исходный код аналога библиотеки string.h

Исходный код 5: библиотека utils.c.

```

1 #include "include/utils.h"
2
3 void memoryCopy(const char *src, int srcPos, char *dest, int destPos, int length) {
4     for (int i = 0; i < length; i++) {
5         dest[destPos + i] = src[srcPos + i];
6     }
7 }
8
9 int getStringLength(const char *string) {
10     int length = 0;
11     while (string[length] != '\0') length++;
12     return length;
13 }

```

Исходный код 6: заголовочный файл include/utils.h.

```

1 #ifndef INCLUDE_UTILS_H
2 #define INCLUDE_UTILS_H
3
4 int getStringLength(const char *);
5 void memoryCopy(const char *, int, char *, int, int);
6
7 #endif //INCLUDE_UTILS_H

```

4.5. Исходный код аналога библиотеки readline.h

Исходный код 7: библиотека input.c.

```
1 #include <stdlib.h>
2 #include <stdio.h>
3
4 #include "include/input.h"
5 #include "include/utills.h"
6 #include "include/output.h"
7
8 char *getString() {
9     char *string = NULL, *buffer = NULL;
10    int ret, bufLength, length = 0;
11
12    do {
13        buffer = malloc((MAX_BUF + 1) * sizeof(char));
14        ret = scanf("%128[^\n]", buffer);
15        if (ret < 0) {
16            free(string);
17            string = NULL;
18        } else if (!ret) {
19            scanf("%*c");
20            if (!string) {
21                printErrorMsg("String cannot be empty, try again: ");
22                ret = 1;
23            }
24        } else {
25            bufLength = getStringLength(buffer);
26            string = realloc(string, length + bufLength + 1);
27            memoryCopy(buffer, 0, string, length, bufLength);
28            length += bufLength;
29        }
30        free(buffer);
31    } while (ret == 1);
32    if (!ret && length > 0) string[length] = '\0';
33
34    return string;
35 }
```

Исходный код 8: заголовочный файл include/input.h.

```
1 #ifndef INCLUDE_INPUT_H
2 #define INCLUDE_INPUT_H
3
4 #define MAX_BUF 128
5
6 char *getString();
7
8 #endif //INCLUDE_INPUT_H
```

4.6. Исходный код вращера потока вывода

Исходный код 9: библиотека output.c.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #include "include/output.h"
5
6 void printDelimiterMsg() {
7     printSuccessMsg("*****\n");
8 }
9
```

```

10 void printExitMsg() {
11     printSuccessMsg("Goodbye...\n\n");
12 }
13
14 void printSuccessMsg(const char *msg) {
15     printf("\033[0;32m");
16     printf("%s", msg);
17     printf("\033[0m");
18 }
19
20 void printErrorMsg(const char *msg) {
21     printf("\033[0;31m");
22     printf("%s", msg);
23     printf("\033[0m");
24 }
25
26 void clearOutput() {
27 #ifdef _WIN32
28     system("cls");
29 #else
30     system("clear");
31 #endif
32 }

```

Исходный код 10: заголовочный файл include/output.h.

```

1 #ifndef INCLUDE_OUTPUT_H
2 #define INCLUDE_OUTPUT_H
3
4 #define INPUT_MSG "Enter any string: "
5
6 void printDelimiterMsg();
7 void printExitMsg();
8 void clearOutput();
9 void printErrorMsg(const char *);
10 void printSuccessMsg(const char *);
11
12 #endif //INCLUDE_OUTPUT_H

```

5. Сравнительный анализ времени

Из сравнения времени выполнения первой и второй программы можно сделать вывод, что для программы с библиотечными функциями требуется больше времени, в сравнении с локальными.

Таблица 1: Анализ времени

| Входная строка | Сдвиг | Время выполнения 1-ой | Время выполнения 2-ой |
|--|-------|-----------------------------|-----------------------------|
| The Tsunami wave crashed against the raised houses and broke the pilings as if they were toothpicks | 24 | 0.001882 | 0.000309 |
| The two walked down the slot canyon oblivious to the sound of thunder in the distance | 56 | 0.000806 | 0.000210 |
| Green should have smelled more tranquil, but somehow it just tasted rotten | 12 | 0.000905 | 0.000222 |
| I always dreamed about being stranded on a desert island until it actually happened | 1 | 0.000430 | 0.000210 |

6. Описание тестовых примеров

Таблица 2: Тестовые примеры

| Входная строка | Сдвиг | Ожидаемое значение s | Полученное значение s |
|---|-------|---|---|
| Hello world! | 9 | oHell wd!lor | oHell wd!lor |
| He had unknowingly taken up sleepwalking as a nighttime hobby | 12 | He had nknowinglyu kenta up sleepwalking as a httimenig bbyho | He had nknowinglyu kenta up sleepwalking as a httimenig bbyho |
| Lucifer was surprised at the amount of life at Death Valley | 34 | rLucife asw edsurpris at het atomnu of feli at hDeat Vylael | He had nknowinglyu kenta up sleepwalking as a httimenig bbyho |
| Here would be an empty shift btw =) | 0 | Here would be an empty shift btw =) | Here would be an empty shift btw =) |
| Richard Paul Astley (born 6 February 1966) is an English singer, songwriter and radio personality | 2 | chardRi ulPa tyesAl orn(b 6 uaryFebr 66)19 is an glishEn nger,si wtireosgnr dan diora rsonalitype | chardRi ulPa tyesAl orn(b 6 uaryFebr 66)19 is an glishEn nger,si wtireosgnr dan diora rsonalitype |

7. Скриншоты

7.1. Первая программа

```

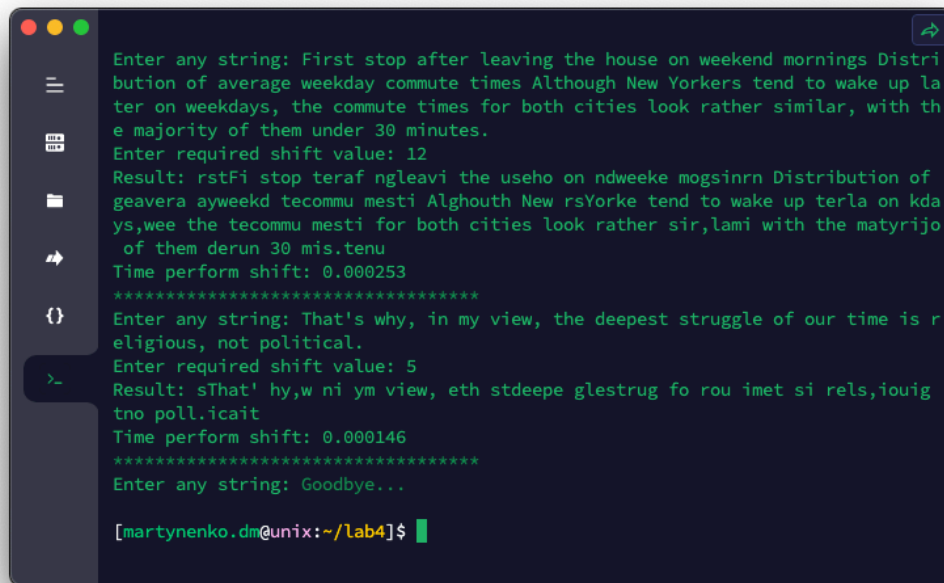
Enter any string: First stop after leaving the house on weekend mornings Distribution of average weekday commute times Although New Yorkers tend to wake up later on weekdays, the commute times for both cities look rather similar, with the majority of them under 30 minutes.
Enter required shift value: 12
Result: rstFi stop teraf ngleavi the useho on ndweeke mogsinrn Distribution of geavera ayweekd tecommu mesti Alghouth New rsYorke tend to wake up terla on kda ys,wee the tecommu mesti for both cities look rather sir,lami with the matyrijo of them derun 30 mis.tenu
Time perform shift: 0.003968
*****
Enter any string: That's why, in my view, the deepest struggle of our time is religious, not political.
Enter required shift value: 5
Result: sThat' hy,w ni ym view, eth stdeepe glestrug fo rou imet si rels,iouig tno poll.icait
Time perform shift: 0.000740
*****
Enter any string:
Goodbye...

[martynenko.dn@unix:~/lab4]$

```

Рис. 9: Работа программы с использованием библиотечных функций.

7.2. Вторая программа

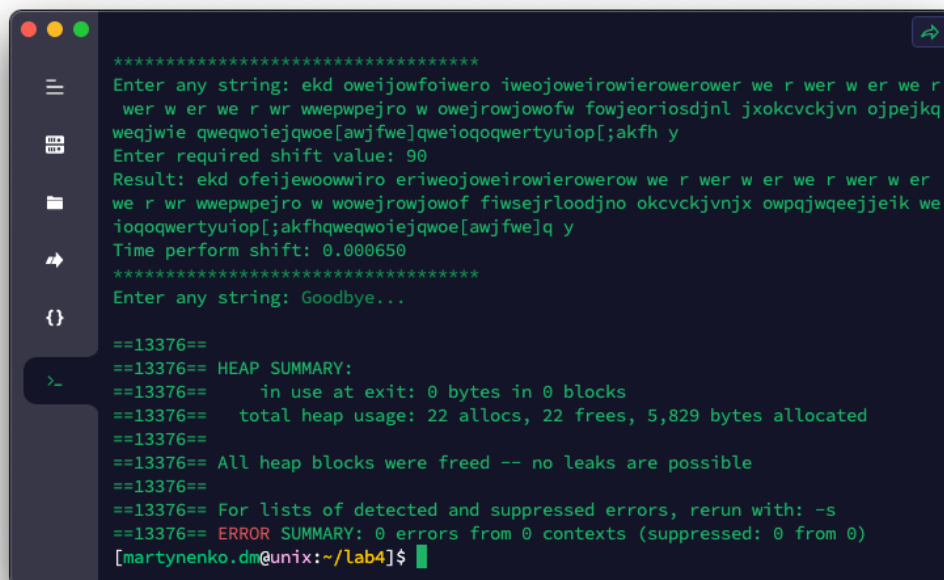


```
Enter any string: First stop after leaving the house on weekend mornings Distribution of average weekday commute times Although New Yorkers tend to wake up later on weekdays, the commute times for both cities look rather similar, with the majority of them under 30 minutes.
Enter required shift value: 12
Result: rstFi stop teraf ngleavi the useho on ndweeke mogsinrn Distribution of geavera ayweekd tecommu mesti Alghouth New rsYorke tend to wake up terla on kdays,wee the tecommu mesti for both cities look rather sir,lami with the matyrjio of them derun 30 mis.tenu
Time perform shift: 0.000253
*****
Enter any string: That's why, in my view, the deepest struggle of our time is religious, not political.
Enter required shift value: 5
Result: sThat' hy,w ni ym view, eth stdeepe glestrug fo rou imet si rels,iouig tno poll.icait
Time perform shift: 0.000146
*****
Enter any string: Goodbye...

[martynenko.dn@unix:~/lab4]$
```

Рис. 10: Работа программы без использования библиотечных функций.

7.3. Использование памяти



```
*****
Enter any string: ekd oweijowfoiwero iweojoweiorowierowerow we r wer w er we r wer w er we r wr wwepwpejro w owejrowjowofw fowjeoriosdjl jxokcvckjvn ojpejkq weqjwie qweqwoiejqwoe[awjfwe]qweioqwertuyiop[;akfh y
Enter required shift value: 90
Result: ekd ofeijewoowwiro eriweojoweiorowierowerow we r wer w er we r wer w er we r wr wwepwpejro w wowejrowjowof fiwsejrlloodjno okcvckjvnjx owpqjwqeejjeik we ioqoqwertyuiop[;akfhqweqwoiejqwoe[awjfwe]q y
Time perform shift: 0.000650
*****
Enter any string: Goodbye...

==13376==
==13376== HEAP SUMMARY:
==13376==   in use at exit: 0 bytes in 0 blocks
==13376==   total heap usage: 22 allocs, 22 frees, 5,829 bytes allocated
==13376==
==13376== All heap blocks were freed -- no leaks are possible
==13376==
==13376== For lists of detected and suppressed errors, rerun with: -s
==13376== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
[martynenko.dn@unix:~/lab4]$
```

Рис. 11: Успешная проверка использования памяти при помощи valgrind.

8. Выводы

В ходе выполнения данной работы были:

1. Изучена работа со строками (`char*`).
2. Изучены библиотеки `readline` и `string`.
3. Созданы свои самописные функции аналоги библиотек.