



Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Робототехники и комплексной автоматизации

КАФЕДРА Системы автоматизированного проектирования (РК-6)

ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ

Студент Гусаров Аркадий Андреевич

Группа РК6-53Б

Тип задания Лабораторная работа №3

Тема лабораторной работы Тренажёр клавиатуры

Студент

подпись, дата

Гусаров А.А.
фамилия, и.о.

Оценка _____

Москва, 2021 г.

Задание на лабораторную работу

Разработать программу тренажера клавиатуры с использованием терминального интерфейса, в которой создается копия образца со случайными пропусками некоторых символов. Пропуски нужно заполнить правильными символами из образца. Длина образца и число пропусков передаются в качестве аргументов командной строки.

Теоретическая часть. Описание алгоритма

На входе алгоритм получает длину строки, которую нужно сгенерировать, и количество пропусков в этой строке. После чего алгоритм генерирует нужную строку и пропуски в ней. В качестве результата дает пользователю возможность отгадать сгенерированную строку и подсчитывает количество совершенных ошибок и время поиска в секундах.

Код программы

```
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <termios.h>
#include <stdio.h>
#include <time.h>

int textmode(int);
int randline(int, int);
int getch();

// Генерируемая строка
static char pattern[80];
// Генерируемая строка со скрытыми символами
static char pattern_hidden[80];
// Функция генерирует случайное число в диапазоне
int get_random_num(int, int);

int get_random_num(int s, int e)
{
    int r = rand() % (e - s);
    return r + s;
}

int textmode(int mode)
```

```

{
    static struct termios con[2];

    if (mode > 0)
        return (tcsetattr(0, TCSAFLUSH, &con[1]));

    tcgetattr(0, &con[0]);
    tcgetattr(0, &con[1]);

    con[0].c_lflag &= ~(ICANON | ECHO | ISIG);
    con[0].c_iflag &= ~(ISTRIP | IXOFF | IXANY | IXON);
    con[0].c_oflag |= CS8;
    con[0].c_cc[VMIN] = 2;
    con[0].c_cc[VTIME] = 1;

    return (tcsetattr(0, TCSAFLUSH, &con[0]));
}

int getch()
{
    unsigned char c[2];
    static int len = 0;

    c[0] = c[1] = 0;

    if ((len = read(0, c, 2)) < 2)
        return (c[0]);
    if (c[0] == 27)
        c[0] = 0;

    ungetc(c[1], stdin);

    return (c[0]);
}

int randline(int len, int skips)
{
    int i = 0;
    /*
    Функция srand() позволяет запускать программу несколько раз
    с различными последовательностями псевдослучайных чисел.
    Функция getpid() возвращает цифровой идентификатор процесса,
    ассоциированного с программой.
    */
    srand(getpid());

    while (i < len)
    {
        // Случайное число от 32 до 126 (коды ASCII)
        pattern[i++] = get_random_num(32, 127);
    }
}

```

```

    }
    return (0);
}

int main(int argc, char *argv[])
{
    const time_t time_start = time(NULL);

    srand(getpid());

    if (argc != 3)
    {
        printf("Wrong number of arguments\n");
        exit(100);
    }

    int i = 0;
    int err = 0;
    unsigned char c;
    int len = atoi(argv[1]);
    int skip = atoi(argv[2]);

    // Формирование случайной строки
    randline(len, skip);

    strcpy(pattern_hidden, pattern);

    // Массив случайных индексов, по которым будут скрыты символы
    for (int j = 0; j < skip; j++)
    {
        int idx = get_random_num(0, len);
        pattern_hidden[idx] = 32;
    }

    // ! write(1, pattern, len);
    write(1, pattern_hidden, len);
    write(1, "\n", 1);
    textmode(0);

    while (i < len)
    {
        switch (c = getch())
        {
            case 0:
                c = '\007';
                switch (getch())
                {
                    case 67:
                        c = pattern[i];
                        break;

```

```

        case 68:
            if (i == 0)
                break;
            i--;
            write(1, "\b", 1);
            continue;
        default:
            break;
    }
    break;
case 27:
    i = len;
    c = '\007';
    break;
default:
    if (c != pattern[i])
        c = '\007';
    break;
}

(c == '\007') ? err++ : i++;
write(1, &c, 1);
}

write(1, "\n", 1);
textmode(1);

const time_t time_end = time(NULL);

printf("Число ошибок: %d\n", err);
printf("Продолжительность ввода (сек): %ld\n", time_end -
time_start);

return (err);
}

```

Результат работы программы

Вызов:

./a.out 2 1

Результат:

$\wedge V$ [

$\wedge VQ$ [

Число ошибок: 45

Продолжительность ввода (сек): 23