ГУАП

КАФЕДРА № 53

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| к.т.н, доц.. |  |  |  | А.В. Туманова |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| Обработка текстовых данных |
| по дисциплине: ОСНОВЫ ПРОГРАММИРОВАНИЯ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4131 |  | 17.03.22 |  | Д.А. Кузнецов |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

1. **Цель работы**

Целью работы является изучение представления строковых данных в языке С и алгоритмов их обработки.

1. **Задание**

Разработать функцию, которая выполняет ту обработку символьной строки, которая определена в Вашем индивидуальном задании. Примечание: строки представляются как массив символов. При реализации функции запрещается пользоваться функциями обработки строк библиотек языка C (например, функцией srtlen стандартной библиотеки C, функциями библиотек "string.h" и "ctype.h").

Примечание: если использовались операторы динамического выделения памяти, то следует вставить дополнительный код, обнаруживающий утечки памяти.



1. **Описание созданных функций**

**Имя:** main

**Назначение**: ввод/вывод начальной и конечной строк, а также ввод данных для преобразования

**Входные данные**: bool: f1, f2 - флаги

bool\*: fpointer1, fpointer2 – указатели на флаги

int: i - счётчик  
 j - счётчик

k - счётчик

plus – общее количество пробелов, которые нужно добавить

mid – среднее количество пробелов, которые нужно добавить

ost – остаточное количество пробелов, которые нужно добавить

spaces – количество пробелов в предложении

words – количество слов в предложении

size – размер предложения (в символах)

expsize – ожидаемый размер предложения (в символах)

char\*: str, outstring – входной и выходной массивы строк соответственно

**Побочный эффект:** отсутствует

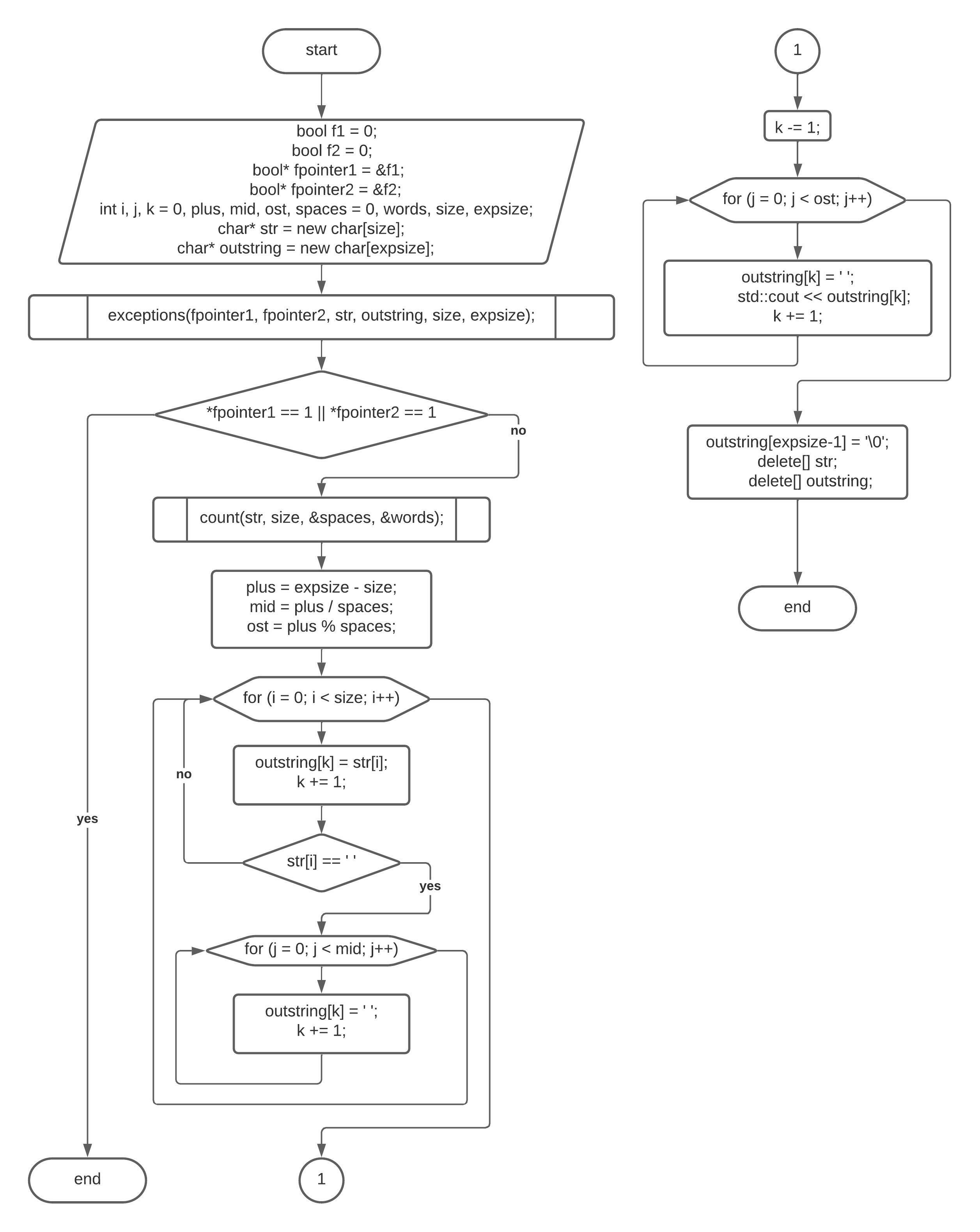
**Тестовые данные:** size = 23

str = ‘What is your name, huh?’

expsize = 60

**Прототип:** int main()

**Блок-схема:**



1. **Текст программы:**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71 | **int** **main**() {  // переменные указатели для функции exceptions  **bool** f1 = **0**;  **bool** f2 = **0**;  **bool**\* fpointer1 = &f1;  **bool**\* fpointer2 = &f2;  setlocale(LC\_ALL, "rus");  **int** i, j, k = **0**, plus, mid, ost, spaces = **0**, words, size, expsize;  std::cout << "Введите длину строки: ";  std::cin >> size;  std::cout << std::endl;  size += **1**; // Дополнительное пространство для нуль-терминатора  // Ссылка на массив слов  **char**\* str = **new** **char**[size];  std::cout << "Введите строку: ";  std::cin.get(); // Добавляем, чтобы сработал метод getline  (из за остаточного \n)  std::cin.getline(str, size);  std::cout << std::endl;  std::cout << "Введите необходимую длину строки: ";  std::cin >> expsize;  **if** (expsize < **0**) {  std::cout << "Длина строки не может быть отрицательной!**\n**";  **return** **0**;  }  expsize += **1**; // Дополнительное пространство для нуль-терминатора  // Ссылка на конечный массив  **char**\* outstring = **new** **char**[expsize];  exccehck(fpointer1, fpointer2, str, outstring, size, expsize);  // Если введённые данные не прошли проверку, то завершаем программу  **if** (\*fpointer1 == **1** || \*fpointer2 == **1**)  **return** **0**;  count(str, size, &spaces, &words);  std::cout << std::endl;  transform(str, outstring, size, expsize, spaces);  std::cout << std::endl;  std::cout << "'" << outstring << "'";  std::cout << std::endl;  std::cout << "Длина строки: ";  std::cout << strlen(outstring);  std::cout << std::endl;  **delete**[] str;  **delete**[] outstring;  // Для обнаружения утечек памяти  \_CrtSetReportMode(\_CRT\_WARN, \_CRTDBG\_MODE\_FILE);  \_CrtSetReportFile(\_CRT\_WARN, \_CRTDBG\_FILE\_STDOUT);  \_CrtSetReportMode(\_CRT\_ERROR, \_CRTDBG\_MODE\_FILE);  \_CrtSetReportFile(\_CRT\_ERROR, \_CRTDBG\_FILE\_STDOUT);  \_CrtSetReportMode(\_CRT\_ASSERT, \_CRTDBG\_MODE\_FILE);  \_CrtSetReportFile(\_CRT\_ASSERT, \_CRTDBG\_FILE\_STDOUT);  \_CrtDumpMemoryLeaks();  } |
|  |  |

**Имя:** count

**Назначение**: подсчёт количества слов и пробелов в предложении

**Входные данные:** char\*: str – входная строка

int: size – размер входной строки, i - счётчик

int\*: spaces, words – количество пробелов и слов соответственно

**Побочный эффект:** отсутствует

**Тестовые данные:** str = ‘What is your name, huh?’

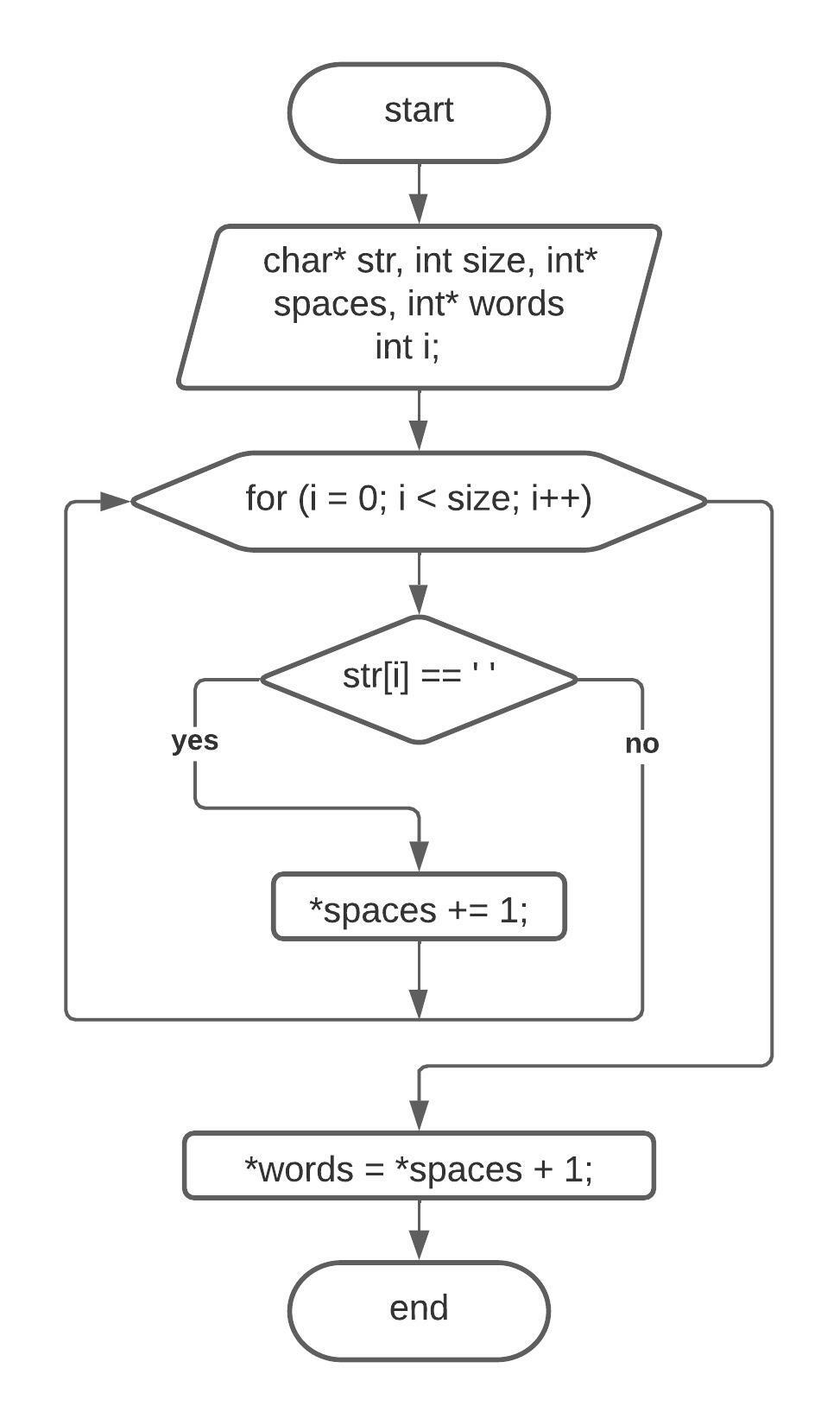
size = 23

\*spaces = 0

\*words = 0

**Прототип:** void count(char\* str, int size, int\* spaces, int\* words)

**Блок-схема:**



1. **Текст программы:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10 | **void** **count**(**char**\* str, **int** size, **int**\* spaces, **int**\* words) {  **int** i;  **for** (i = **0**; i < size; i++) {  **if** (str[i] == ' ') {  \*spaces += **1**;  }  }  \*words = \*spaces + **1**;  } | |  |  |  |

**Имя:** exccheck

**Назначение**: обработка исключений при вводе

**Входные данные:** bool\*: fpointer1, fpointer 2 - указатели

char\*: str, outstring – входная и выходная строки соответственно

int: size, expsize – размеры входной и выходной строки соответственно

**Побочный эффект:** отсутствует

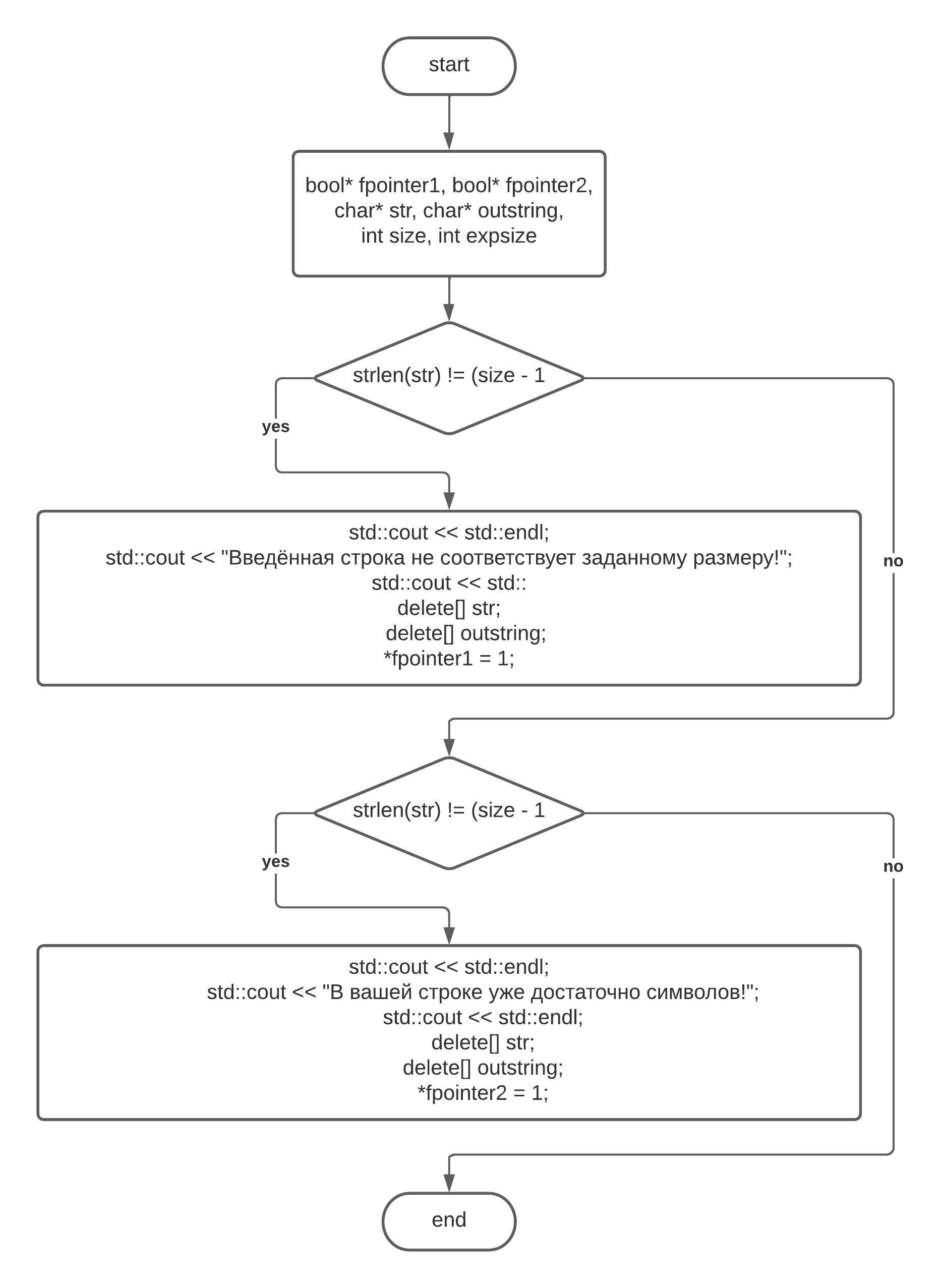
**Тестовые данные:** \*fpointer1 = 0; \*fpointer2 = 0;

str = ‘What is your name, huh?’; outstring = ‘’;

size = 23; expsize = 60;

**Прототип:** void exccehck(bool\* fpointer1, bool\* fpointer2, char\* str, char\* outstring, int size, int expsize)

**Блок-схема:**



1. **Текст программы:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | **void** **exccehck**(**bool**\* fpointer1, **bool**\* fpointer2, **char**\* str, **char**\* outstring,  **int** size, **int** expsize) {  **if** (strlen(str) != (size - **1**)) {  std::cout << std::endl;  std::cout << "Введённая строка не соответствует заданному размеру!";  std::cout << std::endl;  **delete**[] str;  **delete**[] outstring;  \*fpointer1 = **1**;  }  **else** **if** (expsize <= size) {  std::cout << std::endl;  std::cout << "В вашей строке уже достаточно символов!";  std::cout << std::endl;  **delete**[] str;  **delete**[] outstring;  \*fpointer2 = **1**;  }  } | |  |  |  |

**Имя:** transform

**Назначение**: обработка строки в соответствии с вариантом

**Входные данные:** char\*: str, outstring – входная и выходная строки соответственно

int: size, expsize – размеры входной и выходной строки соответственно

int: spaces – количество пробелов в предложении

**Побочный эффект:** отсутствует

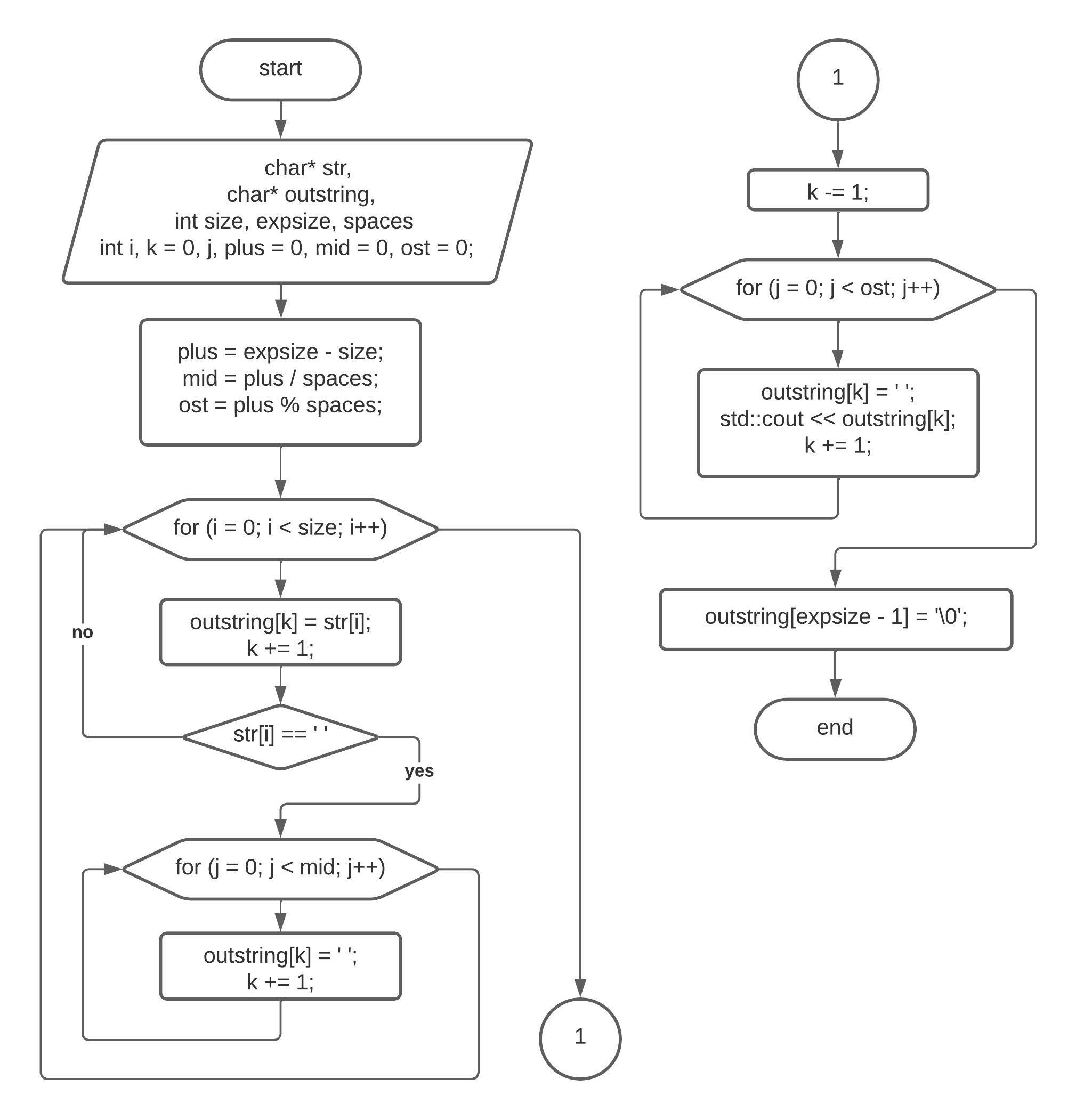
**Тестовые данные:** spaces = 4;

str = ‘What is your name, huh?’; outstring = ‘’;

size = 23; expsize = 60;

**Прототип:** void transform(char\* str, char\* outstring, int size, int expsize, int spaces)

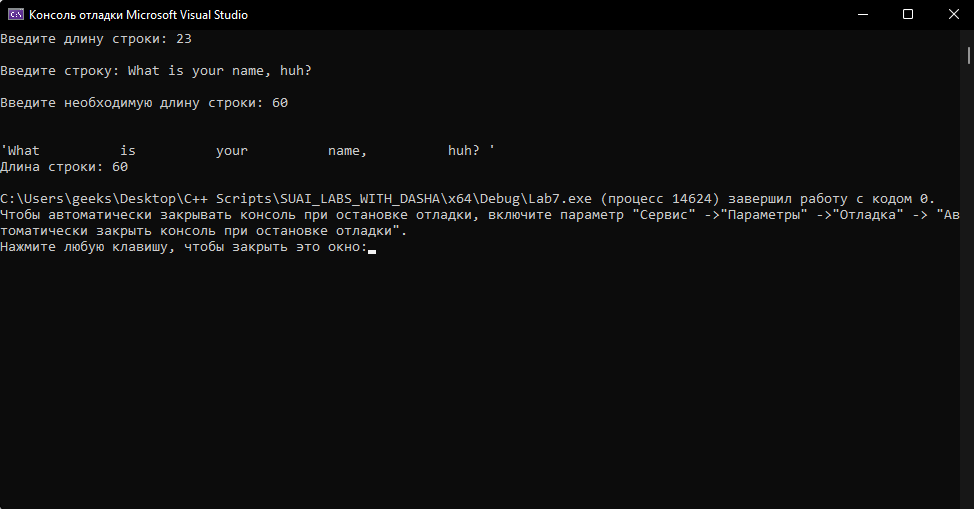
**Блок-схема:**



1. **Текст программы:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32 | **void** **transform**(**char**\* str, **char**\* outstring, **int** size, **int** expsize, **int** spaces) {  **int** i, k = **0**, j, plus = **0**, mid = **0**, ost = **0**;  plus = expsize - size; // Количество пробелов, которые нужно добавить  mid = plus / spaces; // Среднее кол-во пробелов, которое необходимо  добавить  ost = plus % spaces;  // k - счётчик для нового массива  **for** (i = **0**; i < size; i++) {  outstring[k] = str[i];  k += **1**;  **if** (str[i] == ' ') {  **for** (j = **0**; j < mid; j++) {  outstring[k] = ' ';  k += **1**;  }  }  }  // Заполняем остатки пробелов начиная с нуль-терминатора  k -= **1**;  **for** (j = **0**; j < ost; j++) {  outstring[k] = ' ';  std::cout << outstring[k];  k += **1**;  }  // Устанавливаем новый нуль-терминатор  outstring[expsize - **1**] = '\0';  } | |  |  |  |

1. **Пример выполнения программы**

****

1. **Анализ результатов и выводы**

В ходе лабораторной работы я изучил представление строковых данных в языке С и алгоритмы их обработки.