Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Санкт-петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича»

(СПбГУТ)

ОтчЁт  
ЛАБОРАТОРНАЯ РАБОТА №9

**Работа со строками**

Руководитель,  
старший преподаватель Помогалова А. В.

подпись, дата

Исполнитель,  
группа ИКПИ-33 Коньков М. Д.

подпись, дата

УСЛОВИЯ РАБОТЫ

**Вариант № 12:**

Создать программу по вычислению значений функций y и z:

|  |  |
| --- | --- |
| Вариант | Задание |
| 12 | Поменять местами первое и последнее слово в каждой строке. |

**Общая формулировка:**

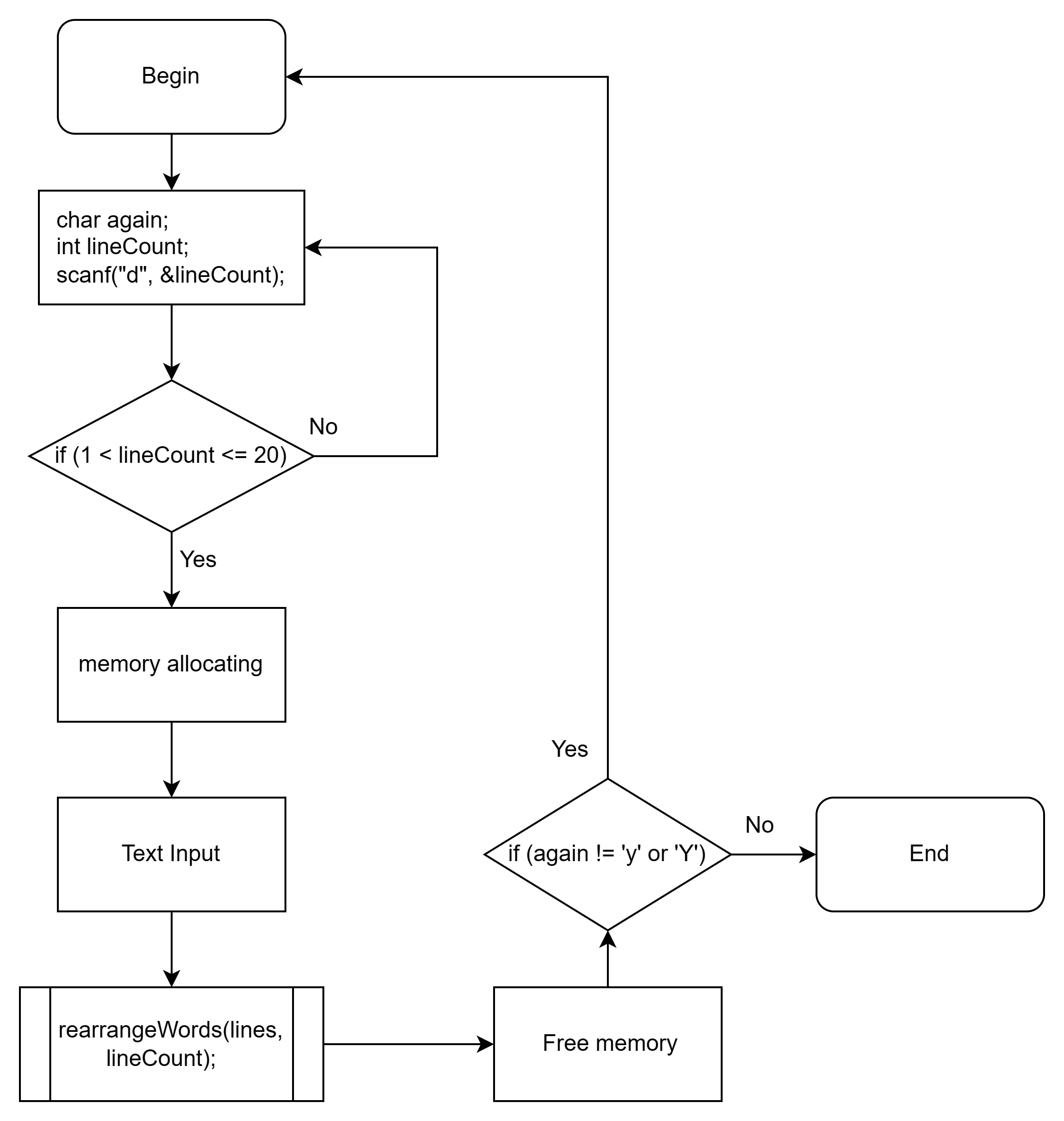
Необходимо считать текст, введённый с клавиатуры по определённым условиям, затем поменять местами первое и последнее слово в каждой строке и вывести изменённый текст.

# Общий алгоритм решения

В данной лабораторной работе необходимо работать со строками. Поделим задачу на 5 подзадач:

1. Просим пользователя указать число строк и затем проверить это число на условие ‘0 < число строк <= 20’;
2. Выделяем память для динамического массива строк, где каждая будет являться массивом с символами (получается работа с двумерным массивом);
3. Просим пользователя поочерёдно ввести текст из слов построчно, после каждой итерации дублируя память для новой строки;
4. Выводим с помощью функции ‘rearrangeWords();’ изменённый текст с заменой первого и последнего слова в каждой из строк, после чего освобождаем память компьютера после вывода текста;
5. Предлагаем пользователю повторить программу или остановить.

## Общий алгоритм решения



|  |  |  |  |
| --- | --- | --- | --- |
| N | Обозначение в задаче | Идентификатор | Назначение |
| 1 | *numCount* | *numCount* | Порядковый номер слова |
| 2 | *n* | *lineCount* | Количество строк текста |
| 3 | *i, j* | *i, j* | Итераторы циклов функции |
| 4 | *again* | *again* | Повторение программы |

## Тестирование

Для тестирования программы выбираем контрольный набор исходных данных: *lineCount = 3; again = 0; inputText(\*):*

*Hey, brother Mario!*

*Do you want to eat pizza tomorrow?*

*Because I will go.*

**Переменные**:

|  |  |  |  |
| --- | --- | --- | --- |
| Переменная | lineCount | again | inputText |
| Рабочий набор | 3 | 0 | \* |

1. Листинг программы lab9.c (C)

#include "stdio.h"

#include "stdlib.h"

#include "string.h"

#include "math.h"

#include "functions.c"

#include "functions.h"

#define MAX\_LENGTH 1000

int main()

{

char again;

while (1)

{

// Enter number of lines

int lineCount;

system("cls");

printf("\033[0;36mEnter count of strings: \033[0m");

scanf("%d", &lineCount);

if ( (lineCount < 1) || (lineCount > 20) )

{

while ( (lineCount < 1) || (lineCount > 20) )

{

printf("\n\033[0;31mError! Count of strings shoud be '0 < x <= 20'.\033[0m\n");

printf("\033[0;36mEnter count of strings: \033[0m");

scanf("%d", &lineCount);

}

}

// Allocating memory for the text

while (getchar() != '\n' && getchar() != EOF) // memory allocating cycle (error preventing)

{

char \*\*lines = (char\*\*)malloc(lineCount \* sizeof(char \*));

}

// Entering original text

char \*lines[MAX\_LENGTH];

printf("\n\033[0;32m--- Starting text ---\033[0m\n");

for (int i = 0; i < lineCount; i++) // text input cycle

{

char inputText[MAX\_LENGTH];

fgets(inputText, sizeof(inputText), stdin);

inputText[strcspn(inputText, "\n")] = 0;

lines[i] = strdup(inputText);

}

// Output fixed text

printf("\n\033[0;32m--- Fixed text ---\033[0m\n");

rearrangeWords(lines, lineCount);

for (int i = 0; i < lineCount; i++) // memory free cycle

{

free(lines[i]);

}

// Repeating program

printf("\n\033[0;33mRepeat program? \033[0m('y'/'Y' --- yes / other --- no)\nAnswer: ");

scanf(" %c", &again);

if ( (again != 'y') && (again != 'Y') )

{

break;

}

}

return 0;

}

1. листинг файла functions.c (C)

#include "stdio.h"

#include "stdlib.h"

#include "string.h"

#define MAX\_LENGTH 1000

void rearrangeWords(char \*\*lines, int lineCount)

{

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

{

char \*line = lines[i];

char \*word; // массив со словами

char \*firstWord[MAX\_LENGTH];

char \*lastWord[MAX\_LENGTH];

char \*goodWords[MAX\_LENGTH];

int numCount = 0;

word = strtok(line, " \n"); // разделить по строкам

firstWord[i] = word;

if (word != NULL)

{

while (word != NULL)

{

goodWords[numCount] = word;

// printf("%s ", goodWords[numCount]);

numCount++;

lastWord[i] = word;

word = strtok(NULL, " \n");

}

}

goodWords[0] = lastWord[i];

goodWords[numCount-1] = firstWord[i];

for (int j = 0; j < numCount; i++)

{

if (goodWords[j] != NULL)

{

printf("%s ", goodWords[j]);

}

}

printf("\n");

}

printf("\n");

}

1. листинг файла functions.h

#ifndef functions\_h

#define functions\_h

void rearrangeWords(char \*\*lines, int lineCount);

#endif

1. Результат работы программы lab9.c (С)

Enter count of strings: -5

Error! Count of strings shoud be '0 < x <= 20'.

Enter count of strings: 21

Error! Count of strings shoud be '0 < x <= 20'.

Enter count of strings: 3

--- Starting text ---

Hey, brother Mario!

Do you want to eat pizza tomorrow?

Because I will go.

--- Fixed text ---

Mario! brother Hey,

tomorrow? you want to eat pizza Do

go. I will Because

Repeat program? ('y'/'Y' --- yes / other --- no)

Answer: no

D:\Univer\Prog\My\9>