|  |  |
| --- | --- |
| Пермский национальный исследовательский политехнический университет  Кафедра информационные технологии и автоматизированные системы | |
|  |  |
|  |  |
| Лабораторная работа №6  "Строки" | |
|  |  |
|  | Выполнил:  Студент группы ИВТ-24-2б  А.С. Ваулин  Проверил:  Доцент кафедры ИТАС  О.А. Полякова |
|  |  |
| Пермь 2025 | |

# Постановка задачи

Задана строка, состоящая из символов. Символы объединяются в слова. Слова друг от друга отделяются одним или несколькими пробелами. В конце текста ставится точка. Текст содержит не более 255 символов. Выполнить ввод строки, используя функцию Getline(s) и обработку строки в соответствии со своим вариантом:

Преобразовать строку так, чтобы все слова в ней стали идентификаторами, слова, состоящие только из цифр - удалить.

# Анализ задачи

**Цель задачи:**

Необходимо разработать программу, которая обрабатывает текстовую строку. Строка содержит слова, разделенные пробелами, и заканчивается точкой. Программа должна преобразовать эту строку таким образом, чтобы каждое слово, если оно не состоит только из цифр, стало соответствовать правилам идентификатора. Слова, состоящие исключительно из цифр, должны быть удалены из итогового текста.

**Входные данные:**

* Текстовая строка, ограниченная длиной в 255 символов.
* Слова в строке разделены одним или несколькими пробелами.
* Строка обязательно завершается точкой.

**Выходные данные:**

* Модифицированная строка, в которой:
  + Все слова, которые не являются чисто цифровыми, преобразованы в идентификаторы.
  + Слова, состоящие только из цифр, исключены из строки.
  + Слова в результирующей строке разделены одиночными пробелами.
  + В конце результирующей строки сохраняется точка.

**Процесс преобразования слова в идентификатор:**

Преобразование слова в идентификатор включает следующие шаги:

1. **Первый символ:** Проверка первого символа слова. Если он не является буквой или символом подчеркивания, он заменяется на символ подчеркивания.
2. **Последующие символы:** Проверка всех остальных символов слова (начиная со второго). Если символ не является буквой, цифрой или символом подчеркивания, он заменяется на символ подчеркивания.

**Общий алгоритм решения задачи:**

1. **Получение входной строки:** Программа должна принять на вход строку текста.
2. **Разделение строки на слова:** необходимо разделить входную строку на отдельные слова. Разделителями между словами выступают один или несколько пробелов. Точка в конце строки также является разделителем или признаком конца текста.
3. **Обработка каждого слова:** для каждого выделенного слова нужно выполнить следующие действия:
   * **Проверка на "цифровое" слово:** определить, состоит ли слово исключительно из цифр.
   * **Удаление цифровых слов:** если слово состоит только из цифр, оно должно быть проигнорировано и не включено в результат.
   * **Преобразование в идентификатор (для нецифровых слов):** если слово не является цифровым, его необходимо преобразовать в соответствии с правилами формирования идентификаторов, описанными выше.
   * **Формирование результирующей строки:** Преобразованные слова (или исходные, если они не были цифровыми и не требовали преобразования) должны быть добавлены к результирующей строке. Между словами в результирующей строке должен быть ровно один пробел.
4. **Добавление точки в конец:** убедиться, что результирующая строка заканчивается точкой.
5. **Вывод результата:** вывести полученную результирующую строку.

# Блок-схема

начало

void processWord(char\* word, char\* ident, char\* result, int& res\_pos)

#include <iostream>

#include <Windows.h>

using namespace std;

const int MAX\_LEN = 1024;

bool number\_word = true;

bool check\_complete = false;

for (int j = 0; word[j] && !check\_complete; ++j)

if (word[j] < '0' || word[j] > '9')

number\_word = false;

check\_complete = true;

if (!number\_word)

α

ζ

{

{

}

1

0

1

0

{

}

1

0

α

int ident\_pos = 0;

bool last\_underscore = false;

if ((word[0] < 'A' || (word[0] > 'Z' && word[0] < 'a') || word[0] > 'z') && word[0] != '\_')

if (ident\_pos < MAX\_LEN - 1)

ident[ident\_pos++] = '\_';

last\_underscore = true;

for (int j = 0; word[j]; ++j)

bool valid = (word[j] >= 'A' && word[j] <= 'Z') ||

(word[j] >= 'a' && word[j] <= 'z') ||

(word[j] >= '0' && word[j] <= '9') ||

word[j] == '\_';

if (valid)

β

γ

δ

ε

{

1

0

{

}

1

0

{

1

0

1

0

β

if (ident\_pos < MAX\_LEN - 1)

ident[ident\_pos++] = word[j];

last\_underscore = (word[j] == '\_');

γ

if (ident\_pos < MAX\_LEN - 1)

ident[ident\_pos++] = '\_';

last\_underscore = true;

if (!last\_underscore)

δ

}

{

}

{

}

1

0

{

}

1

0

{

}

1

0

{

}

ε

ident[ident\_pos] = '\0';

for (int j = 0; ident[j] && res\_pos < MAX\_LEN - 1; ++j)

result[res\_pos++] = ident[j];

if (res\_pos < MAX\_LEN - 1)

result[res\_pos++] = ' ';

ζ

return;

}

}

1

0

1

0

int main()

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

char input[MAX\_LEN] = { 0 };

cout << "Введите строку: ";

cin.getline(input, MAX\_LEN);

char result[MAX\_LEN] = { 0 };

char word[MAX\_LEN] = { 0 };

char ident[MAX\_LEN] = { 0 };

int res\_pos = 0;

int word\_pos = 0;

η

{

η

for (int i = 0; input[i]; ++i)

if (input[i] != ' ' && input[i] != '\t' && input[i] != '\n')

if (word\_pos < MAX\_LEN - 1)

word[word\_pos++] = input[i];

if (word\_pos > 0)

word[word\_pos] = '\0';

processWord(word, ident, result, res\_pos);

word\_pos = 0;

if (word\_pos > 0)

word[word\_pos] = '\0';

processWord(word, ident, result, res\_pos);

θ

1

0

{

}

1

0

{

}

{

}

1

0

1

0

{

}

1

0

{

}

θ

if (res\_pos > 0)

result[res\_pos - 1] = '.';

result[0] = '.';

result[1] = '\0';

cout << "Результат: " << result << endl;

return 0;

}

1

0

{

}

# Код программы

#include <iostream>

#include <Windows.h>

using namespace std;

const int MAX\_LEN = 1024;

// Функция processWord: обрабатывает слово, преобразует в идентификатор, добавляет в результат.

void processWord(char\* word, char\* ident, char\* result, int& res\_pos)

{

bool number\_word = true;

bool check\_complete = false;

// Проверяем, является ли слово числом (состоит только из цифр).

for (int j = 0; word[j] && !check\_complete; ++j)

{

if (word[j] < '0' || word[j] > '9')

{

number\_word = false;

check\_complete = true;

}

}

if (!number\_word)

{

int ident\_pos = 0;

bool last\_underscore = false;

// Если первый символ не буква и не подчеркивание, добавляем подчеркивание в начало идентификатора.

if ((word[0] < 'A' || (word[0] > 'Z' && word[0] < 'a') || word[0] > 'z') && word[0] != '\_')

{

if (ident\_pos < MAX\_LEN - 1) ident[ident\_pos++] = '\_';

last\_underscore = true;

}

for (int j = 0; word[j]; ++j)

{

bool valid = (word[j] >= 'A' && word[j] <= 'Z') ||

(word[j] >= 'a' && word[j] <= 'z') ||

(word[j] >= '0' && word[j] <= '9') ||

word[j] == '\_';

if (valid)

{

if (ident\_pos < MAX\_LEN - 1)

{

ident[ident\_pos++] = word[j];

last\_underscore = (word[j] == '\_');

}

}

// Заменяем недопустимые символы на подчеркивание, избегая повторных подчеркиваний.

else if (!last\_underscore)

{

if (ident\_pos < MAX\_LEN - 1)

{

ident[ident\_pos++] = '\_';

last\_underscore = true;

}

}

}

ident[ident\_pos] = '\0';

for (int j = 0; ident[j] && res\_pos < MAX\_LEN - 1; ++j)

{

result[res\_pos++] = ident[j];

}

if (res\_pos < MAX\_LEN - 1) result[res\_pos++] = ' ';

}

}

int main()

{

SetConsoleCP(1251); // Устанавливаем кодовую страницу ввода для корректного отображения кириллицы.

SetConsoleOutputCP(1251); // Устанавливаем кодовую страницу вывода для корректного отображения кириллицы.

char input[MAX\_LEN] = { 0 };

cout << "Введите строку: ";

cin.getline(input, MAX\_LEN);

char result[MAX\_LEN] = { 0 };

char word[MAX\_LEN] = { 0 };

char ident[MAX\_LEN] = { 0 };

int res\_pos = 0;

int word\_pos = 0;

for (int i = 0; input[i]; ++i)

{

if (input[i] != ' ' && input[i] != '\t' && input[i] != '\n')

{

if (word\_pos < MAX\_LEN - 1)

{

word[word\_pos++] = input[i];

}

}

else

{

// Обрабатываем слово, когда встречаем разделитель.

if (word\_pos > 0) // Проверяем, что слово не пустое.

{

word[word\_pos] = '\0';

processWord(word, ident, result, res\_pos);

word\_pos = 0;

}

}

}

// Обрабатываем последнее слово после окончания цикла (если строка не заканчивается разделителем).

if (word\_pos > 0)

{

word[word\_pos] = '\0';

processWord(word, ident, result, res\_pos);

}

if (res\_pos > 0)

{

result[res\_pos - 1] = '.'; // Заменяем последний пробел на точку.

}

else

{

result[0] = '.'; // Если результат пустой, ставим точку.

result[1] = '\0';

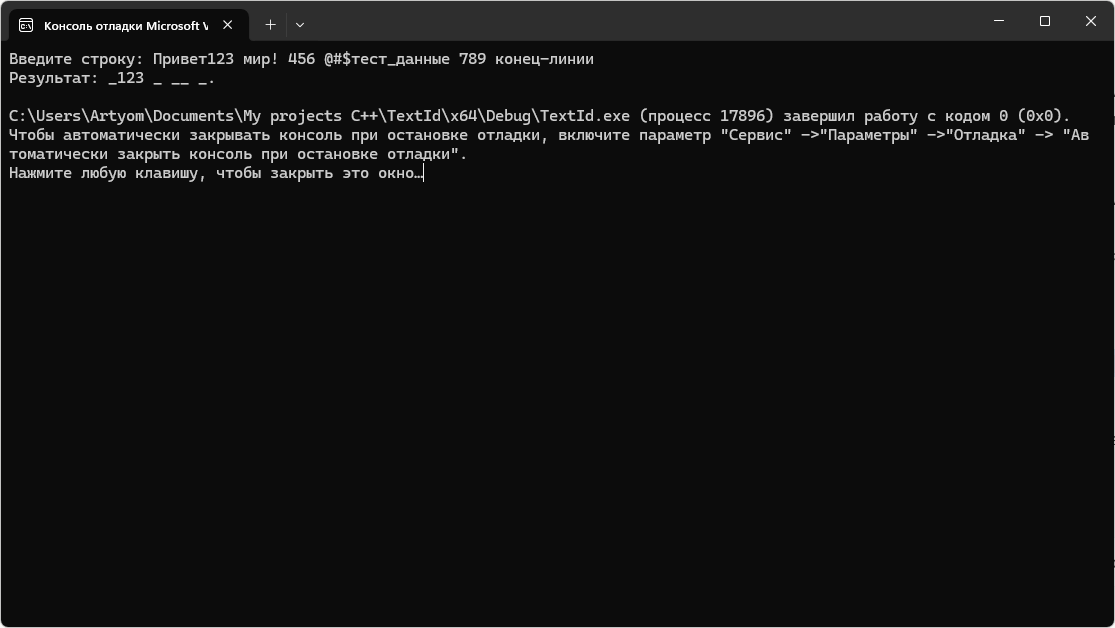
}

cout << "Результат: " << result << endl;

return 0;

}

# Результат



Ссылка на Git: <https://github.com/Artivaa/TextId>