В соответствии с заданием нужно было написать функцию, которая обрезает пробелы в конце переданной ей строки. Функция должна быть написана в расчёте на работу с очень длинными строками с очень большим количеством пробелов, оптимизирована по количеству обращений к памяти.

Самое простое решение «в лоб» было бы просто, начиная с начала массива символов строки C-style посчитать общее число символов перебирая элементы до символа конца строки «\0». А затем вернуться назад, к последнему символу, отличному от пробела, установив символ конца строки сразу, после него. Но при таком решении мы сначала обращаемся к памяти N число раз, где N – это число символов в строке, а затем обращаемся к памяти еще M раз, где M – число пробелов в конце строки.

Альтернативным вариантом решения задачи было бы при первом прохождении строки с начала в конец запоминать положение первого найденного пробела и, если пройдя до конца строки, не обнаружили символов отличных от пробела – обрезать строку сразу за этим индексом. Если же мы встретим далее отличные от пробела символы, то продолжить поиск в массиве следующего начала интервала пробелов и т.д. В этом случае после прохождения строки от начала в конец и и найдя ее длину не нужно возвращаться по массиву для поиска последнего не пробельного символа и число обращений к памяти сокращается до N (числа символов в строке).

Итоговый код функции:

**void TrimRight( char \*s )**

**{**

**//Определяем длину строки без пробелов**

**int length = 0;**

**int index = 0;**

**bool isSpaceLine = false;**

**while (s[length] != '\0')**

**{**

**while (s[length] != '\0' && s[length] != ' ')**

**++length;**

**if (s[length] == '\0')**

**break;**

**if (s[length] == ' ')**

**{**

**index = length;**

**isSpaceLine = true;**

**++length;**

**while (s[length] != '\0' && s[length] == ' ')**

**++length;**

**if (s[length] == '\0')**

**break;**

**if (s[length] != ' ')**

**isSpaceLine = false;**

**}**

**}**

**if (isSpaceLine == true)**

**s[index] = '\0';**

**};**

Для проверки корректности работы функция была протестирована следующими вариантами входных данных:

1. Строка содержащая пробелы в начале, середине и в конце;
2. Строка нулевой длины;
3. Строка не содержащая пробелы;
4. Строка состоящая только из одних пробелов
5. Строка с пробелами в начале.

Тестирование показало корректные результаты.

Код программы содержащий реализацию функции и ее тесты:

#include <iostream>

#include <cstring>

#include <chrono>

//Напишите функцию, которая обрезает пробелы в конце переданной ей строки.

//Функция должна быть написана в расчёте на работу с очень длинными строками

//с очень большим количеством пробелов, оптимизирована по количеству обращений к памяти.

void TrimRight( char \*s )

{

//Определяем длину строки без пробелов

int length = 0;

int index = 0;

bool isSpaceLine = false;

while (s[length] != '\0')

{

while (s[length] != '\0' && s[length] != ' ')

++length;

if (s[length] == '\0')

break;

if (s[length] == ' ')

{

index = length;

isSpaceLine = true;

++length;

while (s[length] != '\0' && s[length] == ' ')

++length;

if (s[length] == '\0')

break;

if (s[length] != ' ')

isSpaceLine = false;

}

}

if (isSpaceLine == true)

s[index] = '\0';

};

void BuildString(char \*mystring, int count)

{

int askiiZeroCode = 48;

int wordLength = 7500;

int index = 0;

for (int i = 0; i < wordLength/10; i++)

{

for (int j = 0; j <= 9; j++)

{

if(j != 0)

mystring[i\*10+j] = askiiZeroCode + j;

else

mystring[i\*10+j] = ' ';

index++;

}

}

for (int i = index; i < count-1; i++)

{

mystring[i] = ' ';

}

mystring[count - 1] = '\0';

};

int main()

{

int count = 10000;

char \*teststring1;

//Случай 1

teststring1 = new char[count];

BuildString(teststring1, count);

std::cout << "Start teststring1: \"" << teststring1 << "\"" << '\n';

std::cout << "Start teststring1 has " << strlen(teststring1) << " letters.\n";

TrimRight(teststring1);

std::cout << "Finish teststring1: \"" << teststring1 << "\"" << '\n';

std::cout << "Finish teststring1 has " << strlen(teststring1) << " letters.\n";

delete[] teststring1;

//Случай 2

char teststring2[] = "";

std::cout << "Start teststring2: \"" << teststring2 << "\"" << '\n';

std::cout << "Start teststring2 has " << strlen(teststring2) << " letters.\n";

TrimRight(teststring2);

std::cout << "Finish teststring2: \"" << teststring2 << "\"" << '\n';

std::cout << "Finish teststring2 has " << strlen(teststring2) << " letters.\n";

//Случай 3

char teststring3[] = "MonolitString";

std::cout << "Start teststring3: \"" << teststring3 << "\"" << '\n';

std::cout << "Start teststring3 has " << strlen(teststring3) << " letters.\n";

TrimRight(teststring3);

std::cout << "Finish teststring3: \"" << teststring3 << "\"" << '\n';

std::cout << "Finish teststring3 has " << strlen(teststring3) << " letters.\n";

//Случай 4

char teststring4[] = " ";

std::cout << "Start teststring4: \"" << teststring4 << "\"" << '\n';

std::cout << "Start teststring4 has " << strlen(teststring4) << " letters.\n";

TrimRight(teststring4);

std::cout << "Finish teststring4: \"" << teststring4 << "\"" << '\n';

std::cout << "Finish teststring4 has " << strlen(teststring4) << " letters.\n";

//Случай 5

char teststring5[] = " End";

std::cout << "Start teststring5: \"" << teststring5 << "\"" << '\n';

std::cout << "Start teststring5 has " << strlen(teststring5) << " letters.\n";

TrimRight(teststring5);

std::cout << "Finish teststring5: \"" << teststring5 << "\"" << '\n';

std::cout << "Finish teststring5 has " << strlen(teststring5) << " letters.\n";

return 0;

}

При реализации использовался компилятор GCC версии 12.2.0 и стандарт языка С++ 17.

void

TrimRight( char \*s )

{

//Определяем длину строки без пробелов

int length = 0;

int index = 0;

bool isSpaceLine = false;

**while** (s[length] != '\0')

{

while (s[length] != '\0' && s[length] != ' ')

++length;

if (s[length] == '\0')

break;

if (s[length] == ' ')

{

index = length;

isSpaceLine = true;

++length;

while (s[length] != '\0' && s[length] == ' ')

++length;

if (s[length] == '\0')

break;

if (s[length] != ' ')

isSpaceLine = false;

}

}

if (isSpaceLine == true)

s[index] = '\0';

};

void TrimRigth\_Alt(char \* s)

{

auto firstSpaceIndex{char};

while (true)

{

switch(\*s)

{

case '\0':

{

break;

}

case ' ':

{

++s;

continue;

}

default:

{

firstSpaceIndex = s;

++s;

continue;

}

}

break;

}

if(\*firstSpaceIndex == ' ')

{

firstSpaceIndex[0] = '\0';

}

else

{

p\_begin\_or\_last\_non\_ws[1] = '\0';

}

return;