/\*

\*\* What I need to do? \*\*

\*\*\* Создать функцию Delete(s,n,l)

\*\*\* удаление из строки s подстроки, начиная с позиции n, длиной l

\*/

/\*

\*\*Explanes of the variables names\*\*

\*\*\* l - длина

\*\*\* n - позиция

\*\*\* s - подстрока

\*\*\* ch - char

\*\*\* i, itter, cc - иттераторы для пробегания по строке/ам

\*\*\* code\_error - Код ошибки для вывода соответствующего сообщения

\*\*\* const\_string - Исходная строка

\*\*\* edited\_string - Измененная строка

\*/

#define N 256

//Переменные

char const\_string[N] = "Supernova will be the first", edited\_string[N], ch;

int i, cc = 0, itter = 0;

long int code\_error;

#include <iostream>

void Error()

{

switch(code\_error)

{

case 0:

std::cout << "Error 0 - Symbols doesn't match!\n"

<< "Please exit and try again";

break;

case 1:

std::cout << "Error 1 - Position is out of range, maybe you entered too long word!\n"

<< "Please exit and try again";

break;

case 2:

std::cout << "Error 2- The word longer than line, or position entered incorrect!\n"

<< "Please exit and try again";

break;

}

}

bool isAllCorrect(char\* s, int n, int l)

{

ch = s[itter];

i = n-1;

while(ch != '\0') // Пока позиция не в конце строки

{

if(

i < n+l // Находится в пределах заданных границ

)

{

if(

ch != const\_string[i++] // Символы не совпадают

)

{

code\_error = 0;

return false; // Ошибка 0

}

if(itter < l) ch = s[++itter];

}

else // Позиция за пределами заданной границы

{

code\_error = 1;

return false; // Ошибка 1

}

if(const\_string[i] == '\0' && ch != 0) // Слово длиннее строки, либо позиция указана не верно

{

code\_error = 2;

return false; // Ошибка 2

}

}

return true;

}

char\* Delete(char\* s, int n, int l)

{

if (!isAllCorrect(s, n, l)) // Ищем проблемы

{

Error(); // Если есть ошибки, то выводим его пользователю

return "\0"; // Возращаем нулевой байт в качестве указания, что что-то пошло не так

}

i = 0; // Обнуление для пробегания по исходной строке

ch = const\_string[i]; // Возвращаем на начало строки, чтобы зайти в while

while(ch != '\0') // Пока позиция не в конце строки

{

if (i < n + l - 1 && i >= n-1) i++; // Если позиция находится в удаляемой зоне, то накручиваем i, как бы пропуская этот промежуток

else

{

ch = const\_string[i++]; // Бежим по строке и берем символы

edited\_string[cc++] = ch; // Записываем символы в новую измененную строку

}

}

return edited\_string;

}

int main(int argc, char const \*argv[])

{

char L[N];

int pos, len;

std::cout << "NO Edited: " << const\_string << "\n";

std::cout << "Enter substring: " << "\n";

std::cin.getline(L, N);

std::cout << "Enter position number: " << "\n";

std::cin >> pos;

std::cout << "Enter the length: " << "\n";

std::cin >> len;

std::cout << "Edited: " << Delete(L, pos, len) << "\n";

std::cin.getline(L, N);

return 0;

}