Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №10**

Дисциплина: основы алгоритмизации и программирования

Тема: Динамические массивы

Выполнил работу

студент группы РИС-20-1б

Ремянников А. В.

Проверила

Доцент кафедры ИТАС

к.т.н Полякова О. А.

Пермь, 2020

**Цель работы**

Организация динамических массивов.

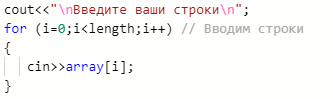
**Постановка задачи**

(Вариант 19) Написать программу, в которой создаются динамические массивы и выполнить их обработку в соответствии со своим вариантом. Сформировать массив строк. Удалить из него строку, начинающуюся и заканчивающуюся на букву "а".

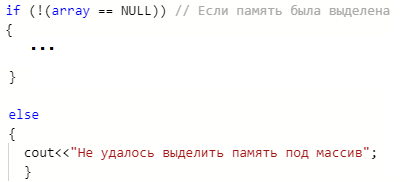
**Анализ задачи**

**1.** Для решения задачи необходимо:

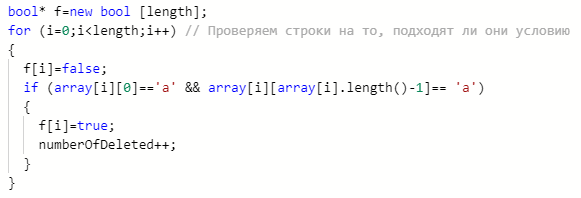
**1.1.** Предоставить пользователю возможность ввести массив строк.



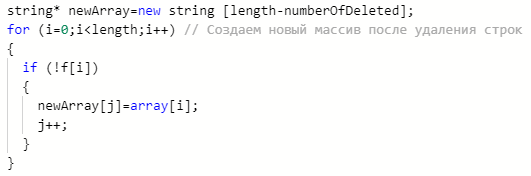
**1.2.** Предусмотреть возможность того, что в памяти не хватит свободного места для создания массива заданной длины и в таком случае сообщить об ошибке.



**1.3.** Создать дополнительный массив f, содержащий информацию о каждой строке (значение true- строка начинается и оканчивается символом 'a', иначе- false).



**1.4.** Создать новый массив и занести туда только строки с индексом, соответствующим значению false массива f.



**2.** В ходе работы были использованы следующие типы данных:

**2.1.** Для экономии памяти и для возможности создавать массив заданной длины, используется оператор new для работы с динамическими массивами.



**2.2.** Для хранения информации о строках в массиве f используется тип bool, поскольку он хранит информацию о том, соответствует ли строка условию “начинается и заканчивается на букву "а".”

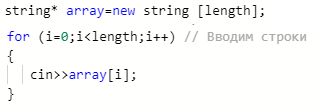


**2.3.** Для хранения самих строк используется динамический массив из string.



**3.** Для решения задачи данные были представлены в следующем виде:

**3.1.** Введенные строки представляются в виде динамического массива из string, где каждый элемент- отдельная строка.



**4.** Для операций ввода и вывода использовались следующие операторы и функции:

**4.1.** Для ввода строк используется оператор cin.

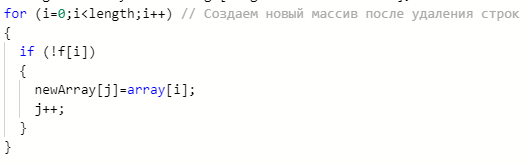


**4.2.** Для вывода измененного массива строк используется оператор cout.

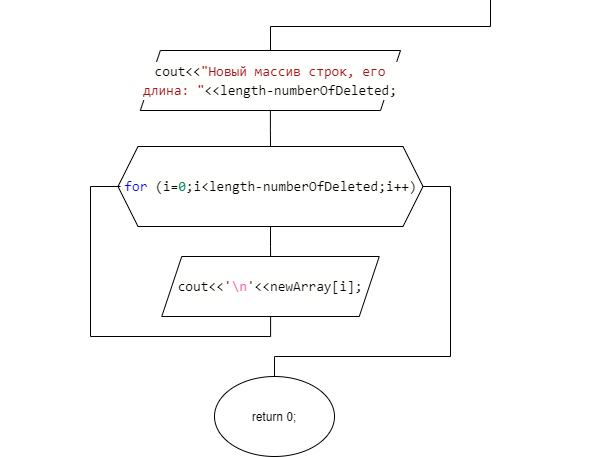
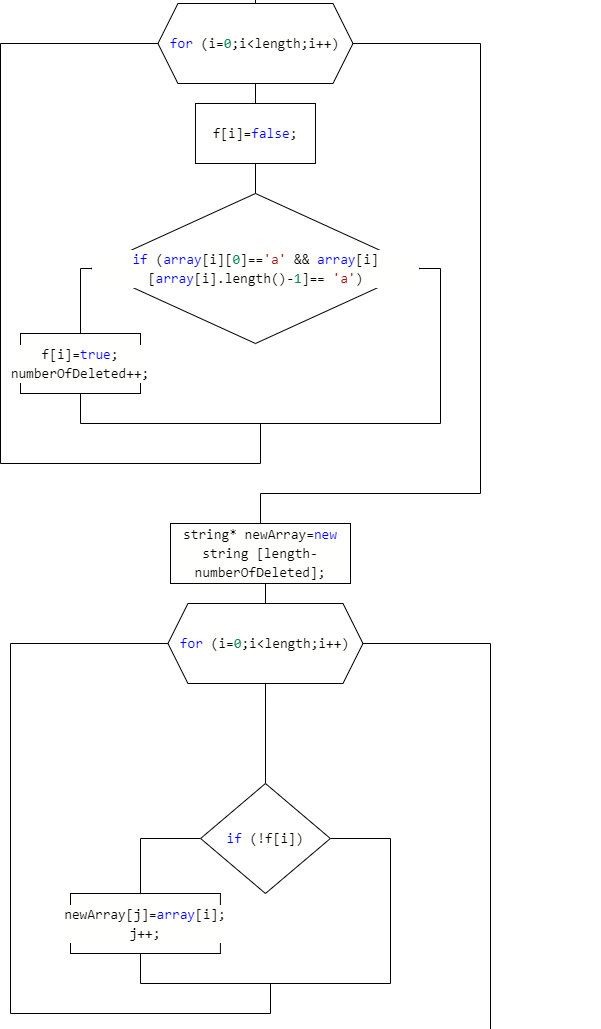
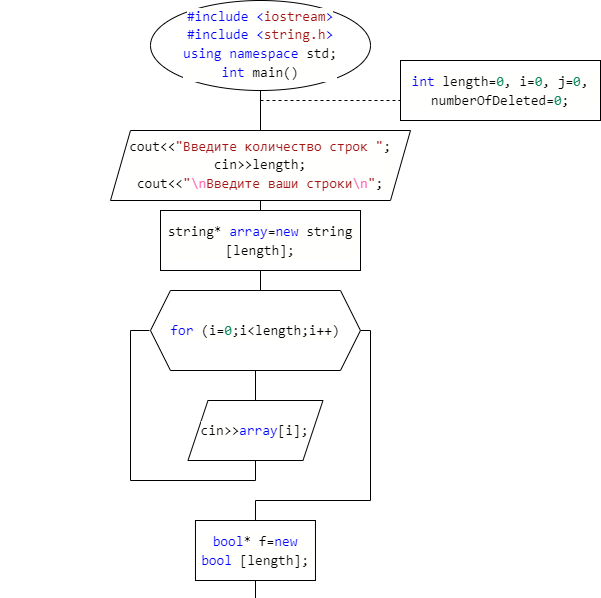


**5.** Поставленные задачи будут решены следующими действиями:

**5.1.** С помощью массива f создается новый массив, в который не входят строки, в начале и конце которых стоит символ ‘а’.



**Блок-схема программы**



**Решение**

#include <iostream>

#include <string.h>

using namespace std;

int main()

{

int length=0, i=0, j=0, numberOfDeleted=0;

cout<<"Введите количество строк ";

cin>>length;

string\* array=new string [length];

if (!(array == NULL)) // Если память была выделена

{

cout<<"\nВведите ваши строки\n";

for (i=0;i<length;i++) // Вводим строки

{

cin>>array[i];

}

bool\* f=new bool [length];

for (i=0;i<length;i++) // Проверяем строки на то, подходят ли они условию

{

f[i]=false;

if (array[i][0]=='a' && array[i][array[i].length()-1]== 'a')

{

f[i]=true;

numberOfDeleted++;

}

}

string\* newArray=new string [length-numberOfDeleted];

for (i=0;i<length;i++) // Создаем новый массив после удаления строк

{

if (!f[i])

{

newArray[j]=array[i];

j++;

}

}

cout<<"Новый массив строк, его длина: "<<length-numberOfDeleted;

for (i=0;i<length-numberOfDeleted;i++)

{

cout<<'\n'<<newArray[i];

}

delete [] array;

delete [] newArray;

}

else

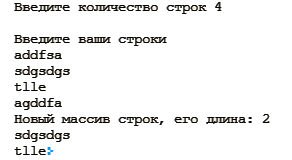
{

cout<<"Не удалось выделить память под массив";

}

}

**Скриншоты результатов работы программы**

****

**GitHub**

<https://github.com/sugarrrfqs/pnrpuLab10/tree/main>