Министерство цифрового развития, связи и массовых коммуникаций РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет

телекоммуникаций и информатики» (СибГУТИ)

Кафедра телекоммуникационных сетей и вычислительных средств

**Курсовая работа**

## **Поиск палиндромов в тексте Вариант 13"**

Выполнил: студент I курса

ИВТ, гр. ИА-132

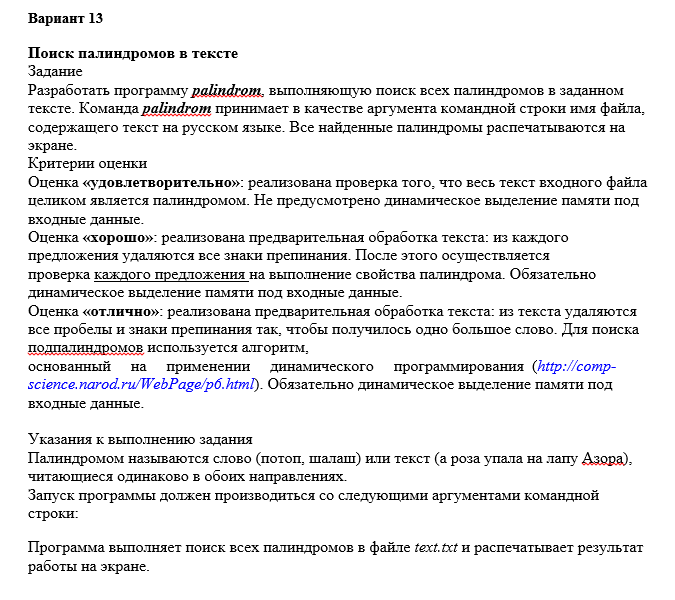
Кулаков К.В.

Проверил:

Вейлер А. И

Новосибирск 2022

Вариант 13



Объяснение алгоритма:

1. Пользователь передает имя файла, программа проверяет правильность введенных данных, и открывает файл на чтение.

В главном цикле:

1. Осуществляется переход к следующему предложению через fseek(файл, количество символов всех предыдущих предложений, от начала)

Цикл для обработки предложений:

Пока символ не будет являться точкой или файл не закончится:

1. Подсчитываем количество символов в предложении (для перемещения)
2. Подсчитываем количество символов кроме точки и переноса на новую строку, для определения длинны массива
3. Подсчитываем количество символов кроме знаков препинания, пробелов и символов для кодирования русских символов (208,209 и т.д по ansii) для определения длинны массива под преобразованное предложение
4. Как только встречается точка или доходим до конца файла, переходим к началу текущего предложения и увеличиваем сумму символов для перехода
5. Объявляем 2 массива для хранения, неизмененного и отредактированного предложений
6. Заполняем каждый массив исключая из них определенные символы
7. Если встречается Заглавные буквы, то мы увеличиваем символ на 32(по таблице ansii), превращая их в строчные

Алгоритм поиска палиндрома:

Палиндром — это слова или предложения, которые читаются в двух направлениях, значит, чтобы проверить является ли слово палиндромом, нужно взять первую половину этого слова и сравнить с перевернутой второй половиной, если они совпадут, то слово является палиндромом

На предыдущих шагах мы обработали предложение и превратили его в одно слово, исключая все ненужные знаки

Т.к. я использую представление символов, в виде таблице asii, то нам достаточно сравнивать попарно числа (1и последнее) (2 и предпоследнее) и т.д.

Если хоть 1 пара не совпадает, то слово не является палиндромом

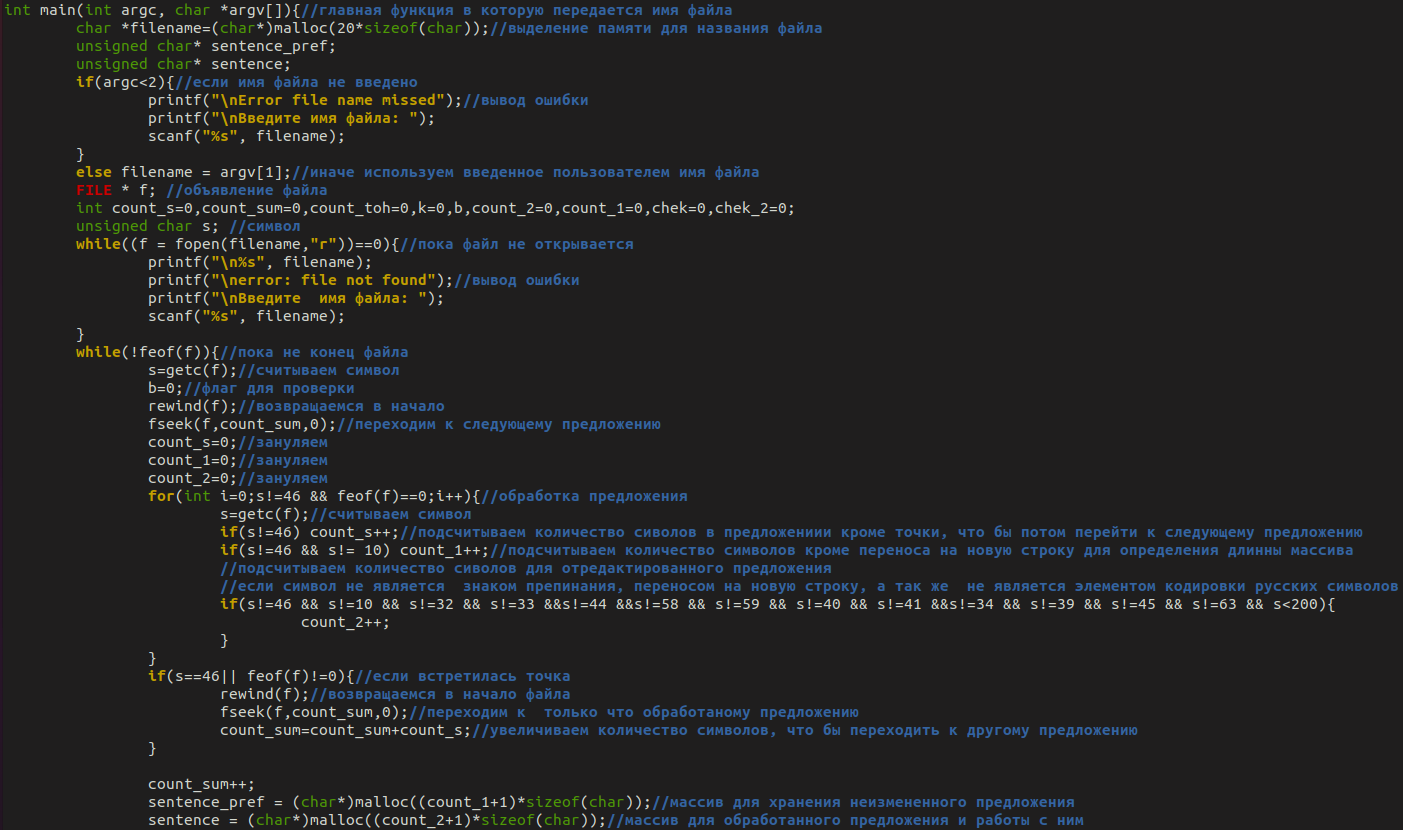
Если цикл пройден, то слово является палиндромом

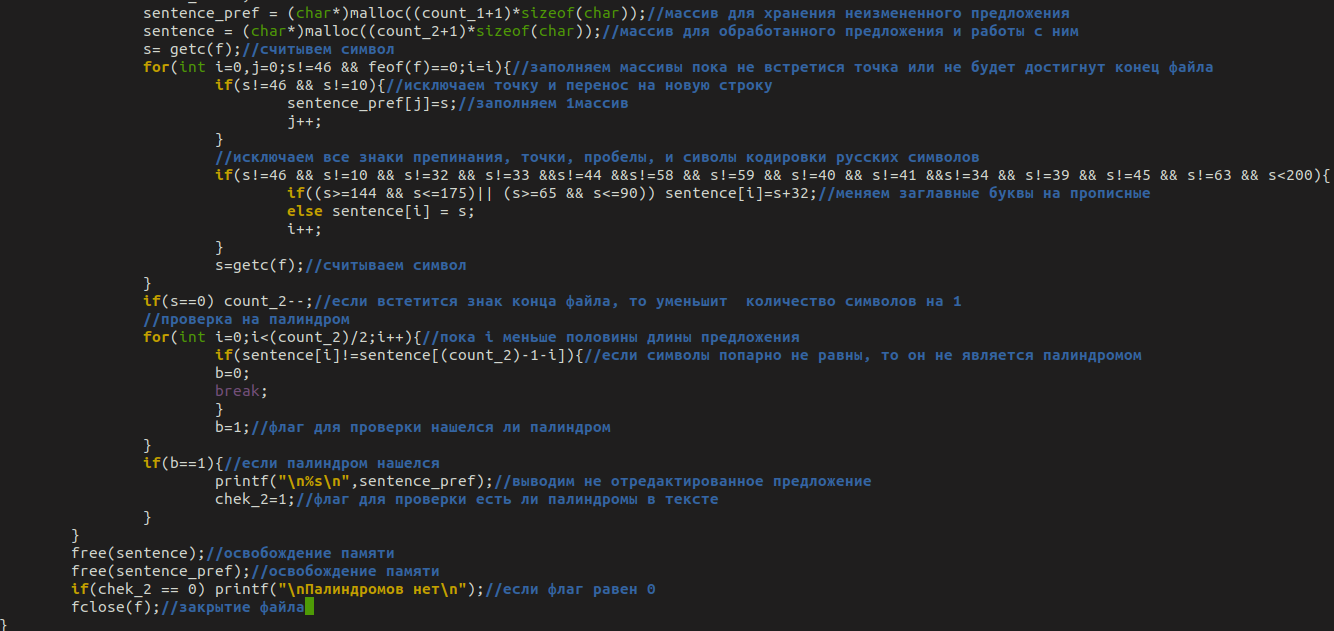
Если палиндром нашелся, то выводим неизмененную версию предложения

Подключение библиотек:

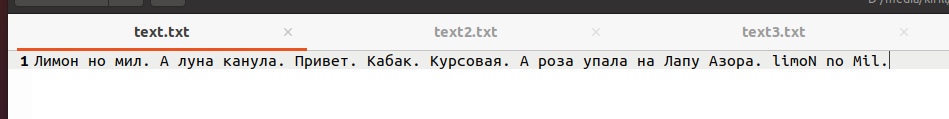


Главная функция:





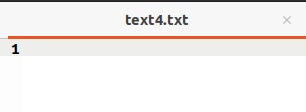
Примеры работы программы:

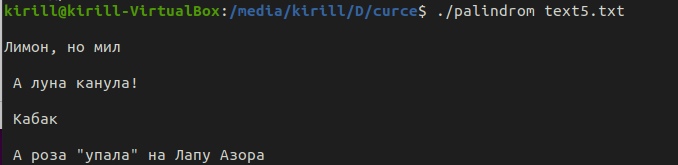
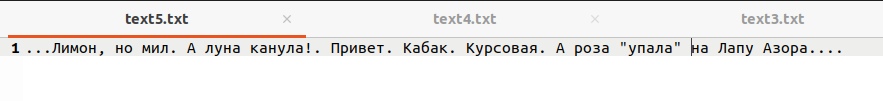






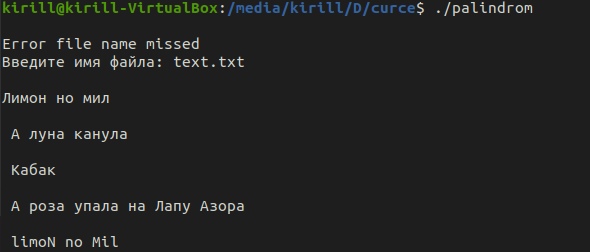
https://sun9-33.userapi.com/s/v1/if2/EtJz9I9zEy__dK1VsGHez0OB0H-U4KRGNrOzXR_MUISIt0iumONvh7oDgZ8UmL5EGEh7GS9T5VHCjRGiUDAcnkKn.jpg?size=631x58&quality=96&type=album

https://sun9-77.userapi.com/s/v1/if2/H0XWk8g8cC88KoIMXpr3XvXd0knRqzgF0KD8paowvDLS0mFFyDQUaQMBzdPU7dTckE8qET4IbrKwA92qMvL6dHpn.jpg?size=643x58&quality=96&type=album

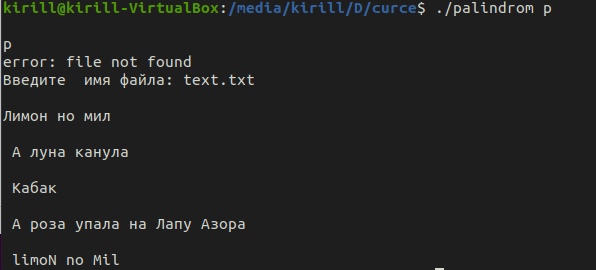


Обработка ошибок:

Если пользователь не передает имя файла:



Если пользователь передает несуществующие название



Листинг программы

Файл palindrom.c

#include <stdio.h> //для ввода вывода

#include <stdlib.h>//для динамической памяти

int main(int argc, char \*argv[]){//главная функция в которую передается имя файла

char \*filename=(char\*)malloc(20\*sizeof(char));//выделение памяти для названия файла

unsigned char\* sentence\_pref;

unsigned char\* sentence;

if(argc<2){//если имя файла не введено

printf("\nError file name missed");//вывод ошибки

printf("\nВведите имя файла: ");

scanf("%s", filename);

}

else filename = argv[1];//иначе используем введенное пользователем имя файла

FILE \* f; //объявление файла

int count\_s=0,count\_sum=0,count\_toh=0,k=0,b,count\_2=0,count\_1=0,chek=0,chek\_2=0;

unsigned char s; //символ

while((f = fopen(filename,"r"))==0){//пока файл не открывается

printf("\n%s", filename);

printf("\nerror: file not found");//вывод ошибки

printf("\nВведите имя файла: ");

scanf("%s", filename);

}

while(!feof(f)){//пока не конец файла

s=getc(f);//считываем символ

b=0;//флаг для проверки

rewind(f);//возвращаемся в начало

fseek(f,count\_sum,0);//переходим к следующему предложению

count\_s=0;//зануляем

count\_1=0;//зануляем

count\_2=0;//зануляем

for(int i=0;s!=46 && feof(f)==0;i++){//обработка предложения

s=getc(f);//считываем символ

if(s!=46) count\_s++;//подсчитываем количество символов в предложении кроме точки, что бы потом перейти к следующему предложению

if(s!=46 && s!= 10) count\_1++;//подсчитываем количество символов кроме переноса на новую строку для определения длинны массива

//подсчитываем количество символов для отредактированного предложения

//если символ не является знаком препинания, переносом на новую строку, а также не является элементом кодировки русских символов

if(s!=46 && s!=10 && s!=32 && s!=33 &&s!=44 &&s!=58 && s!=59 && s!=40 && s!=41 &&s!=34 && s!=39 && s!=45 && s!=63 && s<200){

count\_2++;

}

}

if(s==46|| feof(f)!=0){//если встретилась точка

rewind(f);//возвращаемся в начало файла

fseek(f,count\_sum,0);//переходим к только что обработанному предложению

count\_sum=count\_sum+count\_s;//увеличиваем количество символов, что бы переходить к другому предложению

}

count\_sum++;

sentence\_pref = (char\*)malloc((count\_1+1)\*sizeof(char));//массив для хранения неизмененного предложения

sentence = (char\*)malloc((count\_2+1)\*sizeof(char));//массив для обработанного предложения и работы с ним

s= getc(f);//считываем символ

for(int i=0,j=0;s!=46 && feof(f)==0;i=i){//заполняем массивы пока не встретися точка или не будет достигнут конец файла

if(s!=46 && s!=10){//исключаем точку и перенос на новую строку

sentence\_pref[j]=s;//заполняем 1массив

j++;

}

//исключаем все знаки препинания, точки, пробелы, и сиволы кодировки русских символов

if(s!=46 && s!=10 && s!=32 && s!=33 &&s!=44 &&s!=58 && s!=59 && s!=40 && s!=41 &&s!=34 && s!=39 && s!=45 && s!=63 && s<200){

if((s>=144 && s<=175)|| (s>=65 && s<=90)) sentence[i]=s+32;//меняем заглавные буквы на прописные

else sentence[i] = s;

i++;

}

s=getc(f);//считываем символ

}

if(s==0) count\_2--;//если встетится знак конца файла, то уменьшит количество символов на 1

//проверка на палиндром

for(int i=0;i<(count\_2)/2;i++){//пока i меньше половины длины предложения

if(sentence[i]!=sentence[(count\_2)-1-i]){//если символы попарно не равны, то он не является палиндромом

b=0;

break;

}

b=1;//флаг для проверки нашелся ли палиндром

}

if(b==1){//если палиндром нашелся

printf("\n%s\n",sentence\_pref);//выводим не отредактированное предложение

chek\_2=1;//флаг для проверки есть ли палиндромы в тексте

}

}

free(sentence);//освобождение памяти

free(sentence\_pref);//освобождение памяти

if(chek\_2 == 0) printf("\nПалиндромов нет\n");//если флаг равен 0

fclose(f);//закрытие файла

}