

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Программирование»
Тема: Использование указателей

Студент гр. 1304

Макки К.Ю

Преподаватель

Чайка К.В.

Санкт-Петербург

2021

Цель работы.

Изучение способы работы с текстами используя указатели

Задание.

Напишите программу, которая форматирует некоторый текст и выводит результат на консоль.

На вход программе подается текст, который заканчивается предложением "Dragon flew away!".

Предложение (кроме последнего) может заканчиваться на:

- . (точка)
- ; (точка с запятой)
- ? (вопросительный знак)

Программа должна изменить и вывести текст следующим образом:

- Каждое предложение должно начинаться с новой строки.
- Табуляция (\t, ' ') в начале предложения должна быть удалена.
- Все предложения, в которых есть число 555, должны быть удалены.
- Текст должен заканчиваться фразой "Количество предложений до n и количество предложений после m", где n - количество предложений в изначальном тексте (**без учета** терминального предложения "Dragon flew away!") и m - количество предложений в отформатированном тексте (без учета предложения про количество из данного пункта).

Вариант 4.

Основные теоретические положения.

В данной лабораторной работе использовалась библиотека `stdio.h`, `string.h` и `stdlib.h` способы работы со строками из массивы символов и указатели.

Выполнение работы.

Функция – `punc(int *text, int i)` принимает указатель на массив и индекс элемента и проверяет является ли он знак пунктуации и возвращает 0 или 1, чтобы использовать его как условие

`find(char *text, int i, int len)` –

`*read(int *res)` – функция, которая считывает из терминала текст и в случае не достатка памяти динамически выделяет дополнительное количество памяти.

`free_all(char **s, int len)` – освобождает поэлементно память из массива.

`printlist(char **s, int len)` – печатает на экран обработанный текст по условиям задачи.

Выводы.

В ходе лабораторной работы были изучены способы работы с текстом через указатели на языке Си используя динамические массивы.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

```
#define N 50
#define M 10
#define G 10
#define end "Dragon flew away!"
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

int punc(char *text,int i){
    if (text[i]!='.' && text[i]!='?' && text[i]!=';' && text[i]!
    ='!')
        return 1;
    else
        return 0;
}

int find(char *text,int i,int len){
    int j;
    for (j=i;j<len;j++)
        if (!(punc(text,j)))
            return j-1;
    return j;
}

char *read(int *res){
    int i=0,len=N;
    int n=strlen(end);
    char *text=malloc(sizeof(char)*N);
    char c;
    while (i<n || strncmp(end,text+i-n,n-1)){
        scanf("%c",&c);
        if (c!='\n'){
            text[i]=c;
            if (!(punc(text,i)))
                (*res)++;
            i++;
            if (i>=len-2){
                text=realloc(text,sizeof(char)*(len+N));
                len+=N;
            }
        }
    }
    text[i]='\0';
    return text;
}

void free_all(char **s,int len){
    for (int i=0;i<len;i++)
        free(s[i]);
    free(s);
}
```

```

void printlist(char **s,int len){
    for (int i=0;i<len;i++){
        printf("%s\n",s[i]);
    }
}

int main(){
    char *text=NULL;
    char **sentence=malloc(sizeof(char *)*G);
    int i=0,k=0,j=0,b=0;
    int res=0;
    text=read(&res);
    sentence[0]=malloc(sizeof(char)*M);
    int text_len=strlen(text);
    while (i<text_len){
        if (punc(text,i)){
            if (text[i]=='5' && text[i+1]=='5' && text[i+2]=='5'
&& (!punc(text,i+3) || text[i+3]==' ') && (k==0 || text[i-1]==' ')){
                i=find(text,i,text_len);
                free(sentence[j]);
                b=1;
            }
            else
                sentence[j][k]=text[i];
            k++;
            if (k>=M)
                sentence[j]=realloc(sentence[j],sizeof(char)*(k+2));
            i++;
        }else{
            if (!b){
                sentence[j][k]=text[i];
                sentence[j][k+1]='\0';
                j++;
            }else
                b=0;
            if (j>=G)
                sentence=realloc(sentence,sizeof(char*)*(j+1));
            sentence[j]=malloc(sizeof(char)*M);
            k=0;
            i++;
            while (text[i]=='\t' || text[i]==' ')
                i++;
        }
    }
    printlist(sentence,j);
    printf("Количество предложений до %d и количество предложений после %d\n",res-1,j-1);
    free(text);
    free_all(sentence,j);
}

```

ПРИЛОЖЕНИЕ Б

ТЕСТИРОВАНИЕ

Таблица Б.1 - Примеры тестовых случаев

№ п/п	Входные данные	Выходные данные	Комментарии
1.	Тест на мозвм вар.4	Правильный ответ	Приянт
2.	test test 123 work? work test; testwork. hello world. te55st. test555.Dragon flew away!	test test 123 work? work test; testwork. hello world. te55st. test555. Dragon flew away! Количество предложений до 6 и количество предложений после 6	Верный ответ