

ГЛАВА 5

ОБРАБОТКА СИМВОЛЬНЫХ ДАННЫХ

ВВЕДЕНИЕ

Цель данной работы - получение навыков обработки символьной информации, текстов с использованием стандартных функций языка C, а также с помощью своих собственных функций.

Каждый студент должен решить четыре задачи, начиная с задачи, номер которой вычисляется по формуле:

$NЗ = (4*N)\%63$, где N - порядковый номер студента.

ПРИМЕРЫ РЕШЕНИЯ ЗАДАЧ

Задача А

Ввести последовательность слов и распечатать те из них, перед которыми стоят только лексикографически меньшие, а за ними - только большие слова.

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
#define DL_STR 80 // длина вводимой строки
#define KOL_SL 40 // количество слов в строке

void main()
{
    int i, j, k;
    char str[DL_STR], *sl[KOL_SL], *razd=". ,;:!?";

    vvod: clrscr();
    puts("Введите строку");
    gets(str);
    sl[k=0]=strtok(str, razd); // выделяем первое слово
    while((sl[++k]=strtok(NULL, razd))!=0); // выделяем остальные слова
    puts("\n Список упорядоченных слов : ");
    for(i=0; i<k; i++) // цикл по словам
    {
        for(j=0; j<i; j++)
            if(strcmp(sl[i], sl[j])<0) goto sled_sl;
        for(j=k-1; j>i; j--)
            if(strcmp(sl[i], sl[j])>0) goto sled_sl;
        puts(sl[i]);
    sled_sl:
    }
}
```



```

int j, len, p, k, N;
puts("Какова максимальная длина строки?");
scanf("%d",&N);
getchar();
str=(char*)malloc(N);
puts("Введите строку из слов");
gets(str);
puts("Вот преобразованная строка:\n");
sl=strtok(str, razd);
while(sl) // цикл обработки очередного слова
{
    len=strlen(sl);
    for(j=0; j<len; j++) // ищем в слове первую гласную
        if(strchr(glas, sl[j])!=NULL) break;
    j%=len; // если гласных не было - j=0, чтобы не сдвигать
            // вхолостую слово по кругу
    for(k=0; k<j; k++)
        { // j раз сдвигаем слово на одну букву влево циклически
            b1=sl[0];
            for(p=0; p<len-1; p++)
                sl[p]=sl[p+1];
            sl[p]=b1;
        }
    printf("%s ", sl);
    sl=strtok(NULL, razd); // выделяем очередное слово из исходной строки
}
}

```

Задача D

Составить программу, которая печатает аргументы командной строки в прямом порядке, если первым из них идет опция *-s*, и в обратном порядке, если первый аргумент - опция *-r* (саму опцию не распечатывать).

Например, при вызове `C:\prog -r мне не дают ананасы`
должно быть выдано: `ананасы дают не мне`

```

#include <stdlib.h>
#include <stdio.h>
#include <string.h>

void main(int argc, char *argv[])
{
    if(argc<2)
        {
            puts("Нужны аргументы в командной строке!");
        }
}

```

```

    exit(-1);
}
if(argv[1][0]=='-' && argv[1][1]=='s' && argv[1][2]=='\0')
{ ++argv; // будем печатать аргументы, начиная с третьего
  while (--argc>1)
    printf((argc>2)?"%s ":"%s\n", *++argv);
}
else if(strcmp(argv[1], "-r")==0)
  while (--argc>1)
    printf((argc>2)?"%s ":"%s\n", *(argv+argc));
  else puts("Первым должен быть аргумент -s или -r");
}

```

Задача Е

Построить вертикальную гистограмму числа вхождений каждой строчной латинской буквы в текст.

```

#include <stdio.h>

void main()
{
    int i, j, c, max;
    static int hstgrm[26]; // число вхождений каждой буквы в текст
    char ch, str_gist[53]; // массив - очередная строка в гистограмме

    for(i=0; i<52; i++)
        str_gist[i]=' ';
    str_gist[52]='\0';
    puts("Введите текст, заканчивающийся EOF");

    // определяем вклад каждой буквы в гистограмму
    while ((c=getchar())!=EOF)
        if(c<='z' && c>='a') hstgrm[c-'a']++;

    // находим максимальное число вхождений букв в текст; это и будет
    // количество строк в гистограмме
    max=hstgrm[0];
    for(i=1; i<26; i++)
        if(hstgrm[i]>max) max=hstgrm[i];

    // выводим гистограмму
    for(i=max; i>=1; i--)
    {
        for(j=0; j<26; j++)
            if(hstgrm[j]==i) str_gist[2*j+1]='*';
        puts(str_gist);
    }
}

```

```
for(ch='A'; ch<'Z'; ch++) printf("  %c", ch);  
}
```

ЗАДАНИЕ НА ПРОГРАММИРОВАНИЕ

Задача 1

Написать и протестировать функцию $STRP(str1, str2)$, которая возвращает указатель на первое вхождение символа из строки $str2$ в строку $str1$. Если ни один символ строки $str2$ не входит в строку $str1$, то вернуть NULL.

Задача 2

Ввести предложение, слова в котором разделены пробелами и запятыми. Распечатать это предложение, удалив из него те слова, которые встретились там более одного раза.

Задача 3

Написать и протестировать функцию $DELETE(s1, s2)$, которая удаляет из строки $s1$ все символы, встречающиеся в строке $s2$.

Задача 4↑

Даны две символьные строки, состоящие только из цифр (длина каждой - более 10 символов). Считая, что в этих строках находятся очень длинные целые числа, сформировать третью строку - сумму этих чисел.

Задача 5

Дан произвольный текст. Отредактировать текст так, чтобы:

- а) между словами был ровно один пробел;
- б) предложения в тексте разделялись ровно двумя пробелами.

Задача 6

Написать и протестировать функцию $ESCAPE(str1, str2)$, которая при копировании текста из $str1$ в $str2$ преобразует литеры "новая строка" и "табуляция" в видимые последовательности литер $\backslash n$ и $\backslash t$. Сделать также функцию, выполняющую обратное преобразование.

Задача 7

Ввести два предложения и распечатать самые длинные слова, общие для этих предложений. Если нужных слов нет - сообщить об этом.

Задача 8

Если первый аргумент командной строки - опция $-a$, то распечатать остальные аргументы без их первых символов, а если первой идет опция $-r$, то

распечатать аргументы через один в обратном порядке. (Если аргументов нет - выдать сообщение).

Задача 9↑

Произвести выравнивание по правому краю введенного текста, для чего к каждой строке применить функцию $WIDE(str, k)$, которая равномерно вставляет пробелы между словами так, чтобы длина строки str стала равной k . (Величина k должна быть больше длины самой длинной строки текста).

Задача 10

Написать и протестировать функцию $I_TO_B(n, s, b)$, которая переводит целое число n в строку s , представляющую число в системе счисления с основанием b .

Задача 11

Составьте программу, которая реверсирует каждое слово строки str .

Задача 12

Ввести строку, состоящую только из цифр и букв. Распечатать те группы цифр, в которых цифра '7' встречается не более двух раз. (Группа цифр - это последовательность цифр, обрамленная буквами).

Задача 13

Написать и протестировать функцию $STRP(str1, str2)$, которая возвращает указатель на последнее вхождение символа из строки $str2$ в строку $str1$. Если ни один символ строки $str2$ не входит в строку $str1$, то вернуть NULL.

Задача 14

Ввести предложение, слова в котором разделены пробелами и запятыми. Распечатать те слова, которые являются обращениями других слов в этом предложении. Если нужных слов нет - сообщить об этом.

Задача 15

Написать и протестировать аналог функции $STRCSPN()$.

Задача 16

Дан массив символьных строк. Если в командной строке не заданы аргументы, распечатать все строки, а если задана опция $-n$ - распечатать последние n строк.

Задача 17

Распечатать те пары слов, расстояние между которыми наименьшее. (Расстояние между словами - это количество позиций, в которых слова

различаются. Например, расстояние между словами *МАМА* и *ПАПА* или *МЫШКА* и *КОШКА* равно двум).

Задача 18

Написать и протестировать функцию *NXT_BLNK(str, pos)*, которая анализирует строку *str*, начиная с позиции *pos* и возвращает номер первого найденного пробела. Если пробела нет, возвращается 0; если $pos < 0$ или больше длины строки, то возвращается -1.

Задача 19

Дан произвольный текст. Напечатать в алфавитном порядке все буквы, которые входят в этот текст по одному разу.

Задача 20

Распечатать введенное предложение, удалив из него те слова, которые состоят менее чем из трех букв.

Задача 21

Зашифровать текст следующим образом: записать его в матрицу по строкам, а затем переписать по спирали от центра. Прочитать зашифрованный текст.

Задача 22

Распечатать введенные слова, отличные от последнего, преобразовав их следующим образом:

- перенести последнюю букву в начало слова;
- оставить в слове только первые вхождения каждой буквы.

Задача 23

Если первый аргумент командной строки - опция *-d*, то распечатать введенный далее текст без входящих в него цифр, а если указана опция *-r*, распечатать введенный далее текст так, чтобы цифры шли в обратном порядке. В остальных случаях распечатать текст таким, как он был введен.

Задача 24

Написать и протестировать рекурсивную функцию *REVERSE(str)*, которая переворачивает данную строку на том же самом месте. Сравнить время ее работы и время работы нерекурсивной версии.

Задача 25

Написать и протестировать функцию *STRP(str1, str2)*, которая определяет, встретился ли в строке *str1* какой-нибудь символ из строки *str2*. Функция

должна возвращать номера позиций первых совпадающих символов или -1, если совпадений нет.

Задача 26

Из введенного текста распечатать все слова наименьшей длины.

Задача 27↑

Функция Φ преобразования текста определяется следующим образом:

$$\Phi(str) = \begin{cases} \Phi(\gamma)\beta, & \text{если } str = \beta; \gamma \text{ и } \beta \text{ не содержит ';' ,} \\ str, & \text{если в } str \text{ нет ';' .} \end{cases}$$

Реализовать функцию Φ с помощью рекурсивной процедуры.

Задача 28

Распечатать строку, которая получается из введенной строки следующим образом: каждая цифра заменяется на заключенную в круглые скобки последовательность литер '+' (если цифра четная) или '-' (если цифра нечетная), длина которой равна числу, изображаемому цифрой.

Задача 29

Распечатать введенную строку, удалив из неё слова с нечетными номерами и перевернув слова с четными номерами. Например, из строки

во что бы то ни стало

должно получиться

отч от олатс

Задача 30↑

Написать и протестировать аналог функции *STR TOK()*.

Задача 31

Написать и протестировать функцию *ISSUBSTR(str1, str2)*, которая выясняет, является ли строка *str1* подстрокой строки *str2*. Функция должна возвращать номер позиции, с которой начинается подстрока, либо -1, если подстрока не найдена.

Задача 32

Распечатать введенную строку, исключив из нее те символы, которые находятся между скобками '(' ')'. Сами скобки не удалять. Если хотя бы одной скобки нет - сообщить об этом.

Задача 33↑

Если в командной строке заданы опции *-x -n <образец>*, то необходимо распечатать те строки введенного текста с их номерами, в которых не найден

указанный образец, а при указании опций *-x <образец>* распечатать те же строки, только без номеров.

Задача 34

Распечатать те слова, в которых либо буквы упорядочены по алфавиту, либо каждая буква входит в слово не менее двух раз (т.е. слова типа *BEER*, *ABBA*).

Задача 35

Написать и протестировать рекурсивную функцию *STOI(n, str)*, которая преобразует строку десятичных цифр в целое число.

Задача 36

Во введенном тексте найти самое длинное симметричное слово.

Задача 37

Ввести строку, в которую могут входить только цифры и буквы. Распечатать те группы букв, в которых 'A' встречается не менее двух раз. (Группа букв - это последовательность букв, обрамленная цифрами).

Задача 38

Составить частотный словарь вводимого текста. Распечатать его по алфавиту, а справа от каждого слова - частоту, с которой оно встретилось.

Задача 39

Написать и протестировать функцию *ISSUBSTR(str1, str2)*, которая выясняет, является ли строка *str1* подстрокой строки *str2*. Функция должна возвращать указатель на начало подстроки, либо NULL, если подстрока не найдена.

Задача 40↑

Дано целочисленное арифметическое выражение, записанное как строка, в десятичной системе счисления. Проверить правильность записи и вычислить значение этого выражения. Выражение записывается без скобок, операции выполняются в порядке их следования.

Задача 41

Во введенном слове подсчитать количество различных пар букв. (Например, в слове *babacabacd* 5 различных пар букв.)

Задача 42

Написать и протестировать функцию *STREND(str1, str2)*, которая возвращает 1, если строка *str1* расположена в конце строки *str2*, и 0 - в противном случае.

Задача 43

Ввести произвольный текст. Вычислить среднее число слов в предложении и среднюю длину предложения.

Задача 44

Определить, является ли введённая строка правильной записью целого десятичного числа без знака.

Задача 45↑

Если в командной строке заданы опции *-x -n <образец>*, то необходимо распечатать те строки введенного текста с их номерами, в которых указанный образец найден, а при указании опций *-x <образец>* распечатать те же строки, только без номеров.

Задача 46

Проверить, имеется ли в заданном тексте баланс открывающих и закрывающих круглых скобок.

Задача 47

Написать и протестировать аналог функции *STRNCPY()*.

Задача 48

Составить частотный словарь вводимого текста. Слова вместе с их частотами распечатать в порядке убывания частот.

Задача 49

Написать и протестировать аналог функции *STRNCMP()*.

Задача 50

Распечатать в порядке, обратном алфавитному, все буквы, которые входят в текст не менее трёх раз.

Задача 51

Ввести произвольный текст. Вычислить среднюю длину слов в тексте и среднее расстояние (в позициях) между знаками препинания.

Задача 52↑

Определить, является ли введенная строка символов правильной записью формулы. Формула имеет следующий вид:

$\langle \text{формула} \rangle ::= \langle \text{цифра} \rangle \mid (\langle \text{формула} \rangle \langle \text{знак} \rangle \langle \text{формула} \rangle)$
 $\langle \text{знак} \rangle ::= + \mid - \mid *$
 $\langle \text{цифра} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

Задача 53

Написать и протестировать аналог функции *STRNCAT()*.

Задача 54

Распечатать, начиная с последнего, перевернутые слова введенного предложения.

Задача 55

Выделить из строки *str1* все слова, начинающиеся с гласной буквы, а из строки *str2* - слова начинающиеся с согласной. Образовать строку *str3*, состоящую из выделенных слов обеих строк. В новой строке слова должны разделяться двумя пробелами.

Задача 56

Ввести строку, в которую могут входить только цифры и буквы. Распечатать те группы букв, в которых 'А' встречается не менее двух раз. (Группа букв - это последовательность букв, обрамленная цифрами).

Задача 57

Если в командной строке задана опция *-u*, то вводимые в нижнем регистре буквы преобразовывать в буквы верхнего регистра. При указании опции *-l* осуществлять обратное преобразование.

Задача 58

Из введенного текста распечатать только те слова, которые симметричны (*RADAR*, *ANNA*), либо в которых буквы упорядочены в порядке, обратном алфавитному (*TOK*, *ZONA*).

Задача 59

Написать и протестировать функцию *STRS(str1, str2)*, которая определяет, встретился ли в строке *str1* какой-нибудь символ из строки *str2*.

Задача 60

Написать и протестировать функцию *NXT_BLNK(str, pos)*, которая анализирует строку *str*, начиная с позиции *pos* и возвращает указатель на первый найденный пробел. Если пробела нет, либо *pos*<0 или больше длины строки, то возвращается NULL.

Задача 61

Зашифровать текст методом Гронсфельда. Ключом является конечная последовательность цифр, которую записывают подряд над символами шифруемого текста. Цифра, стоящая над литерой, является величиной смещения (т.е. говорит о том, на сколько надо продвинуться вперед по таблице кодировки от текущего символа, чтобы получить для него замену).

Протестировать написанную программу.

Задача 62

Вывести вертикальную гистограмму длин слов введенного текста.

Задача 63

Во введенном тексте указать слово, в котором доля гласных (а, е, о) максимальна.

Задача 64

Написать и протестировать аналог функции *STRSPN()*.