

СИМВОЛЬНЫЕ СТРОКИ. ДИНАМИЧЕСКОЕ РАСПРЕДЕЛЕНИЕ ПАМЯТИ

Цель работы: изучить особенности работы со строковыми объектами как одномерными и двумерными символьными массивами, научиться использовать динамическое распределение памяти.

Примеры решений:

1. Ввести предложение и слова для поиска в предложении.

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

void main()
{
    char x[100], y[50], *ptr;

    printf("Введите строку: ");
    gets(x);
    printf("Введите слова для поиска в строке: ");
    scanf("%s", y);

    ptr = strstr(x,y);
    if(!ptr)
        printf("нет такого слова в предложении\n");
    else
        printf("есть такого слово в предложении\n");
}
```

2. Найти количество гласных букв в введенном предложении.

```
#include <stdio.h>
//получает строку (содержащую гласные буквы) и //символ. Возвращает -1, если символ
не гласная //буква и неотрицательное число, если буква //гласная.
int find(char*, char);

void main()
{
    char str[80];
    gets(str);
    //строка, которая содержит гласные
    char gl[7] = "aouiey";

    for(int i=0,k=0; str[i] != '\0' ;i++)
        if(find(gl,str[i])>=0) k++;

    printf("\n%d",k);
}

//функция поиска заданного символа в строке
int find(char* s, char c)
{
    for (int i=0; s[i] ;i++)
        if(s[i] == c) return i;
    return -1;
}
```

3. Ввести строку, которая содержит целые числа, и преобразовать ее в число.

```
#include<stdio.h>

void main()
{
    char m[100];
    int n = 0,i;
```

```

printf("Ввести строку: ");
scanf("%s",m);

for(i=0; m[i] ;i++)
{
    n *= 10;
    n += m[i]-'0';
}

printf("\n%d\n",n);
}

```

Предположим в строке m записаны следующие символы '1' '2' '4' '5'. Если мы целочисленной переменной n присвоим первый символ, т.е. $m[0]$, то в переменную n запишется код символа '1', т.е. число 49. Если же мы хотим, чтобы в целочисленной переменной n было записано число 1, а не код символа '1', то необходимо от самого символа отнять символ 0, т.е. $n = m[0] - '0'$. И, наоборот, если число необходимо преобразовать в символ, то к числу прибавляется символ 0.

4. Ввести массив из пяти строк, рассортировать в алфавитном порядке

```

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

void main()
{
    char string[5][20],buf[20];
    int i,j,k;

    // ввод массива строк
    . . . . .

    for(i=0;i<4;i++){
        for(k=i,j=i+1;j<5;j++)
            if(strcmp(string[k],string[j])>0)
                k=j;
        strcpy(buf,string[i]);
        strcpy(string[i],string[k]);
        strcpy(string[k],buf);
    }

    puts("");
    //вывод отсортированного массива строк
    . . . . .
}

```

5. Программа, которая динамически выделяет массив целых чисел. Удаление числа из массива и добавление чисел в массив.

```

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

void vvod_mas(double *, int, int);
void vivod_mas(double *, int);
int del_elm(double *, int);

void main(){
    double *arr, *ptr;
    int n, k, nl,s;

    printf("Enter size of array: ");
    scanf("%d",&n);
    if(n <= 0) {
        puts("Errors");
        return;
    }
    arr=(double *) malloc ( n * sizeof(double));
}

```

```

    if( !arr )
    {
        printf("Not enough memory \n");
        exit(1);
    }

    vvod_mas(arr, 0, n);

    printf("\nEnter\n1-to delete\n2- to add\n");
    scanf("%d",&s);

    if(s == 1)
    {
        k = del_elm ( arr,  n);
    if(k == 0)
        printf("There are no such elements\n");
    else{
        if( n == k ) {
            puts("Array is empty");
            free (arr);
            return;
        }
        ptr=(double*)realloc(arr, (n- k)*sizeof( double));
        if( ptr )
        {
            arr = ptr;
            n -= k;
        }
    }
}
else
{
    printf("Enter the numbers to add: ");
    scanf("%d", &n1);
    if(n1 > 0)
    {
        ptr=(double*)realloc(arr, (n+n1)*sizeof(double));
        if( ptr )
        {
            arr = ptr;
            n += n1;
            vvod_mas(arr, n - n1, n);
        }
    }
}

vivod_mas(arr, n);

free(arr);
}

int del_elm( double *a,  int n1 )
{
    int k = 0, i, j;
    double y;

    printf("Vvedite chislo: ");
    scanf("%lf", &y);
    for ( i = 0; i <  n1 - k;  i++)
        if(a[i] == y)
        {
            k++;
            for(j = i; j < n1 - k; j++)
                a[j] = a[j+1];
            i--;
        }
}

```

```

        return k;
    }

void vvod_mas( double *p, int k1, int k2 )
{
    for ( int i=k1;i<k2;i++)
    {
        printf("[%d]= ",i);
        scanf("%lf", &p[i]);
    }
}

void vivod_mas( double *p, int k)
{
    for( int i=0; i < k; i++)
        printf("[%d]=%.2lf\n", i, *(p++));
}

```

6. Двумерный массив. Дописать освобождение памяти.

```

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

void main()
{
    int**arr;
    int row, col, i, j;

    printf("Enter row: ");
    scanf("%d", &row);
    printf("Enter col: ");
    scanf("%d", &col);
    if(row <= 0 || col <= 0)
    {
        puts("Errors");
        return;
    }

    // массив указателей
    arr=(int**)malloc(row*sizeof(int*));

    //одномерный массив
    for(i = 0 ; i < row; i++)
        arr[i]=(int*)malloc(col* sizeof(int));

    if( !arr )
    {
        printf("Not enough memory \n");
        exit(1);
    }

    for(i = 0; i < row; i++)
        for(j = 0; j < col; j++)
        {
            printf("[%d][%d]= ",i,j);
            scanf("%d", &arr[i][j]);
        }

    for(i = 0; i < row; i++)
    {
        for(j = 0; j < col; j++)
            printf("%d\t",arr[i][j]);

        puts("");
    }
}

```

```
//добавить освобождение выделенной памяти  
}
```

Задачи для аудиторной и самостоятельной работы

Задача 1. Ввести строку и распечатать каждый ее символ с новой строки.

Задача 2. Ввести строку и переписать ее в обратном порядке в новую строку.

Задача 3. Ввести строку и заменить символ «a» на символ «!» в данной строке.

Задача 4. Посчитать, сколько раз в словах встречается буква «a» и поменять первый и последний символы.

Задача 5. Посчитать число слов в введенном предложении. Учесть, что предложение может начинаться и заканчиваться пробелами, а также между словами может быть больше, чем один пробел.

Задача 6. Написать функцию присвоения строк.

Прототип функции **char* prsv(char*,char*);** т.е. присвоить вторую строку первой и вернуть первую строку.

Задача 7. Написать функцию добавления строк.

Прототип функции **char* dobav(char*,char*);** т.е. добавить вторую строку первой и вернуть первую строку.

Задача 8. Написать функцию, которая считает количество введенных символов. Прототип функции **int* dlina(char*);** получает строку и возвращает количество введенных символов.

Задача 9. Написать функцию, которая сравнивает строки и возвращает: **0**, если строки одинаковы; **1**, если в первой строке встретился символ код, которого больше чем, код символа во второй строке и **-1** в противном случае. Прототип функции **int srav(char*, char*);**

Задача 10. Имеем 4 строки, например **char x[20], y[20], z[20], t[80];** В первую строку вводим фамилию, во вторую имя, а в третью отчество и используя функции из библиотеки **<string.h>** записываем в строку **z** сначала фамилию, потом пробел, имя, пробел и отчество, пробел. Распечатываем только строку **z**.

Задача 11. Введите строку. Если длина строки а) больше 10, то удалить два последних символа; б) меньше 10, то удалить два первых символа; в) равно 10, то удалить символ посередине.

Задача 12. Введите предложение. Определите, сколько раз в данном предложении встречается введенное слово.

Задача 13. Ввести число и распечатать те цифры данного числа, которые делятся на 2.

Задача 14. Ввести число и посчитать произведение цифр данного числа.

Задача 15. Ввести число и распечатать цифры данного числа через два пробела. Например, если ввели 123456, то печатает 1 2 3 4 5 6.

Задача 16. Ввести дату в строку следующим образом: 12/05/1956 и программа распечатывает строку следующим образом: 12 мая, 1956 года (использовать оператор **switch**).

Задача 17. Ввести число, например 3451, программа должна напечатать три тысячи четыреста пятьдесят один.

Задача 18. Ввести число, посчитать сумму цифр введенного числа, используя **char ***.

Задача 19. За один просмотр исходного текста определить, сколько раз встречается каждый символ.

Задача 20. Ввести массив строк, в который записываются целые числа, преобразовать их в массив чисел и вывести массив чисел.

Задача 21. Дана действительная квадратная матрица порядка N (матрица выделяется динамически). Найти сумму и произведение элементов, расположенных ниже главной диагонали.

Задача 22. Задачу 12 из предыдущего дня переделать следующим образом:

1. Массив выделить динамически.
2. Удаление заданного числа из массива (переопределить размер массива).
3. Добавление чисел в массив (переопределить размер массива).

Задача 23. Ввести строку символов. Определить, является ли данная строка палиндромом (т.е. справа налево и слева направо читается одинаково).