

Ввод-вывод

Прико П.А.

форматированный вывод - printf

%[флаги][ширина][.точность][длина]спецификатор

```
printf("%s\n", string);
```

```
printf("a = %d, b = %d\n", a, b);
```

форматированный вывод – printf

| Спецификатор | Что хотим вывести |
|--------------|---|
| d или i | Целое со знаком в десятичном виде |
| u | Целое без знака в десятичном виде |
| o | Беззнаковое в восьмеричном виде |
| x | Беззнаковое целое в шестнадцатеричном виде |
| X | Беззнаковое целое в шестнадцатеричном виде, верхний регистр |
| f или F | Число с плавающей точкой |
| e | Экспоненциальная форма для числа с плавающей точкой |
| E | Экспоненциальная форма для числа с плавающей точкой, верхний регистр |
| g | Кратчайшее из представлений форматов f и e |
| G | Кратчайшее из представлений форматов F и E |
| a | Шестнадцатеричное представление числа с плавающей точкой |
| A | Шестнадцатеричное представление числа с плавающей точкой, верхний регистр |
| c | Буква |
| s | Строка (нуль-терминированный массив букв) |
| p | Адрес указателя |

форматированный ввод - scanf

%[*][ширина][длинна]спецификатор

```
scanf("%d:%d:%d", &year, &month, &day);
```

форматированный ввод – scanf

| Спецификатор | Описание | Выбранные символы |
|--------------|--------------------------|---|
| i, u | Целые | Произвольное число цифр (0-9), возможно, начинающихся с + или -. Если число начинается с 0, то считывается в восьмеричном формате, если с 0x, то в шестнадцатеричном. |
| d | Десятичное целое | Произвольное число цифр (0-9), возможно, начинающихся с + или -. |
| o | восьмеричное целое | Произвольное число цифр (0-7), возможно, начинающихся с + или -. |
| x | Шестнадцатеричное целое | Произвольное число цифр (0-F), возможно, начинающихся с + или - и префикса 0x или 0X. |
| f, e, g | Число с плавающей точкой | Число, состоящее из набора цифр 0-9, возможно с десятичным разделителем (точкой). Возможно также представление в экспоненциальной форме. C99 позволяет также вводить число в шестнадцатеричном формате. |
| a | | |
| c | Символ | Если ширина не передана, то считывает один символ. Если ширина передана, то считывает нужное количество символов и размещает их в массиве БЕЗ терминального символа на конце. |
| s | Строка | Считывает все не пробельные символы. Если указана ширина, то не более n символов. Ставит на место n+1 символа терминальный. |
| p | Адрес указателя | Последовательность символов, трактуемая как адрес указателя. Формат зависит от реализации, но совпадает с тем, как выводит printf с ключом p |
| [символы] | Множество символов | Считывает только те символы, которые записаны в квадратных скобках, C99 |
| [^символы] | Множество символов | Считывает только те символы, которые не указаны в квадратных скобках, C99 |
| n | Ничего не считывает | Сохраняет число уже считанных символов по указанному адресу |

файловый ввод/вывод: алгоритм

1. Открыть файл с исходными данными
2. Ввести данные
3. Закрыть файл с исходными данными
4. Выполнить действия над данными
5. Открыть файл для вывода
6. Вывести данные в файл
7. Закрыть файл для вывода

файловый ввод/вывод

```
#include <stdio.h>
int main(){
    char simbol;
    int number = 0;
    FILE *f_in; //указатель на файл ввода
    FILE *f_out; //указатель на файл вывода
    int retcode = 100;
    f_in = fopen("text.rtf", "r"); //открываем файл ввода для чтения
    if (f_in == NULL){
        printf(" There is no such file\n");
        return 1;
    }
    for (int counter = 0; counter < 100; counter++){
        fscanf(f_in,"%c",&simbol); //читаем первые 100 символов
        if (simbol == 'e'){
            number = number + 1;
        }
    }
    fclose(f_in); //закрываем файл ввода
    f_out = fopen("out.txt", "w"); //открываем (если его нет - создаем) файл вывода для записи
    fprintf(f_out,"There are %d letters e ", number); //пишем
    fclose(f_out); //закрываем файл вывода
    return 0;
}
```

права доступа к файлу

| Тип | Описание |
|-----|--|
| r | Чтение. Файл должен существовать. |
| w | Запись нового файла. Если файл с таким именем уже существует, то его содержимое будет потеряно. |
| a | Запись в конец файла. Операции позиционирования (fseek, fsetpos, frewind) игнорируются. Файл создаётся, если не существовал. |
| r+ | Чтение и обновление. Можно как читать, так и писать. Файл должен существовать. |
| w+ | Запись и обновление. Создаётся новый файл. Если файл с таким именем уже существует, то его содержимое будет потеряно. Можно как писать, так и читать. |
| a+ | Запись в конец и обновление. Операции позиционирования работают только для чтения, для записи игнорируются. Если файл не существовал, то будет создан новый. |

Спасибо за внимание!