

文件

1. 单选

- 缺省状态下，系统的标准输入文件设备是：_____。
 - A) 键盘
 - B) 显示器
 - C) 软盘
 - D) 硬盘
- C 语言中可以处理的文件类型为：_____。
 - A) 文本文件和数据文件
 - B) 文本文件和二进制文件
 - C) 二进制文件和数据文件
 - D) 以上都不对
- 以下关于文件的叙述中正确的是：_____。
 - A) 对文件操作必须先关闭文件
 - B) 对文件操作必须先打开文件
 - C) 对文件操作顺序没有统一的规定
 - D) 以上都不对
- 在进行文件操作时，写文件的一般含义是：_____。
 - A) 将计算机内存中的信息写入磁盘
 - B) 将磁盘中的信息存入计算机内存
 - C) 将计算机 CPU 中的信息写入磁盘
 - D) 将磁盘中的信息存入计算机 CPU
- 在进行文件操作时，文件读操作的一般含义是：_____。
 - A) 将计算机内存中的信息写入磁盘
 - B) 将磁盘中的信息读入计算机内存
 - C) 将计算机 CPU 中的信息写入磁盘
 - D) 将磁盘中的信息读入计算机 CPU
- C 语言中，从计算机内存中将数据写入文件中，称为：_____。
 - A) 输入
 - B) 输出
 - C) 修改
 - D) 追加
- C 语言中，将文件中的数据读入计算机内存中，称为：_____。
 - A) 输入
 - B) 输出
 - C) 修改
 - D) 追加
- 在 C 语言中，对文件操作的一般步骤是：_____。
 - A) 打开文件，操作文件，关闭文件
 - B) 操作文件，修改文件，关闭文件

- C) 读/写文件，打开文件，关闭文件
- D) 读文件，写文件，关闭文件
- 在下列语句中，将 `c` 定义成文件型指针的是：_____。
 - A) `FILE c;`
 - B) `FILE *c;`
 - C) `file c;`
 - D) `file *c`
- 以读/写方式打开一个已有的文本文件 `file1`，并且已定义 `FILE *fp`。下面 `fopen` 函数正确的调用方式是：_____。
 - A) `fp=fopen("file1", "r")`
 - B) `fp=fopen("file1", "r+")`
 - C) `fp=fopen("file1", "rb")`
 - D) `fp=fopen("file1", "w")`
- 若要打开 A 盘上 `user` 子目录下名为 `abc.txt` 的文本文件进行读、写操作，下面符合些要求的函数调用是：_____。
 - A) `fp=fopen("A:\user\abc.txt", "r")`
 - B) `fp=fopen("A:\user\abc.txt", "r+")`
 - C) `fp=fopen("A:\user\abc.txt", "rb")`
 - D) `fp=fopen("A:\user\abc.txt", "w")`
- 打开一个新的二进制文件，若要用 `fopen` 函数，使该文件要既能读又能写，则文件打开时的文件使用模式是：_____。
 - A) `"rb+"`
 - B) `"wb+"`
 - C) `"b+"`
 - D) `"ab"`
- 当执行 `fopen` 函数时发生错误，则函数的返回值是：_____。
 - A) -1
 - B) TRUE
 - C) 0
 - D) 1
- 当顺利执行了文件关闭操作时，`fclose` 函数的返回值是：_____。
 - A) -1
 - B) TRUE
 - C) 0
 - D) 1
- 若 `fp` 是指向某文件的指针，且已读到此文件末尾（文件尾标识已被读出），则库函数 `feof(fp)` 的返回值是：_____。
 - A) EOF
 - B) 0
 - C) 非零值
 - D) NULL
- 使用 `fgetc` 函数，则打开文件的方式是：_____。
 - A) 只写
 - B) 添加

- C) 读或读/写
- D) B 和 C 都正确
- 当调用 `fputc` 函数输出字符成功，其返回值是：_____。
 - A) EOF
 - B) 1
 - C) 0
 - D) 输出的字符
- 标准库函数 `fgets(s,n,f)` 的功能是：_____。
 - A) 从文件 `f` 中读取长度为 `n` 的字符串存入指针 `s` 所指的内存
 - B) 从文件 `f` 中读取长度不超过 `n-1` 的字符串存入指针 `s` 所指的内存
 - C) 从文件 `f` 中读取 `n` 个字符串存入指针 `s` 所指的内存
 - D) 从文件 `f` 中读取长度为 `n-1` 的字符串存入指针 `s` 所指的内存
- 标准库函数 `fputs(p1, p2)` 的功能是：_____。
 - A) 从 `p1` 指向的文件中读一个字符串存入 `p2` 指向的内存
 - B) 从 `p2` 指向的文件中读一个字符串存入 `p1` 指向的内存
 - C) 从 `p1` 指向的内存中读一个字符串写到 `p2` 指向的文件中
 - D) 从 `p2` 指向的内存中读一个字符串写到 `p1` 指向的文件中
- 若定义 `int a[5]`，`fp` 是指向某一已经正确打开了的文件的指针，以下函数调用中不正确的是：_____。
 - A) `fread(a[0], sizeof(int), 5, fp)`
 - B) `fread(&a[0], 5*sizeof(int), 1, fp)`
 - C) `fread(a, sizeof(int), 5, fp)`
 - D) `fread(a, 5*sizeof(int), 1, fp)`
- `fread(buf, 64, 2, fp)` 的功能是：_____。
 - A) 从 `fp` 文件流中读出整数 64，并存放在 `buf` 中
 - B) 从 `fp` 文件流中读出整数 64 和 2，并存放在 `buf` 中
 - C) 从 `fp` 文件流中读出 64 字节的字符，并存放在 `buf` 中
 - D) 从 `fp` 文件流中读出 2 个 64 字节的字符，并存放在 `buf` 中
- 已知函数的调用形式是 `fread(buffer, size, count, fp)`，其中 `buffer` 代表的是：_____。
 - A) 一个整型变量，代表要读入的数据项总数
 - B) 一个文件指针，指向要读的文件
 - C) 一个指针，指向要读入数据的存放地址
 - D) 一个存储区，存放要读的数据项
- 在 C 程序中，可把整数以二进制形式存放到文件中的函数是：_____。
 - A) `fprintf` 函数
 - B) `fread` 函数
 - C) `fwrite` 函数
 - D) `fputc` 函数
- `fwrite` 函数的一般格式是：_____。
 - A) `fwrite(buffer, count, size, fp)`
 - B) `fwrite(fp, size, count, buffer)`
 - C) `fwrite(fp, count, size, buffer)`
 - D) `fwrite(buffer, size, count, fp)`
- 设有以下结构类型：

```

struct student
{
    char name[8];
    int no;
    float score[4];
}st[50];

```

并且结构体数组 st 中的元素都已有值，若要将这些元素写到文件 fp 中，以下错误的格式是：_____。

- A) fwrite(st, sizeof(struct student), 50, fp)
- B) fwrite(st, 50*sizeof(struct student), 1, fp)
- C) fwrite(st, 50*sizeof(struct student), 25, fp)
- D) for(i=0; i<50; i++) fwrite(st, sizeof(struct student), 1, fp)
- fscanf 函数的正确调用格式是：_____。
 - A) fscanf(文件格式, 格式字符串, 输出列表)
 - B) fscanf(格式字符串, 输出列表, 文件指针)
 - C) fscanf(格式字符串, 文件指针, 输出列表)
 - D) fscanf(文件指针, 格式字符串, 输入列表)
- 以下叙述中错误的是：_____。
 - A) 二进制文件打开后可以先读文件的末尾，而顺序文件不可以
 - B) 在程序结束时，应当用 fclose 函数关闭已打开的文件
 - C) 利用 fread 函数从二进制文件中读数据时，可以用数组名给数组中所有元素读入数据
 - D) 不可以用 FILE 定义指向二进制文件的文件指针
- 下面的程序执行后，文件 test.dat 中的内容是：_____。

```

#include <stdio.h>
void fun(char *fname,char *st)
{
    FILE *myf;
    int i;
    myf=fopen(fname, "w");
    //for(i=0;i<strlen(st);i++)
    for(i=0;*(st+i)!='\0';i++)
        fputc(st[i],myf);
    fclose(myf);
}
int main()
{
    fun("test.dat", "new world");
    fun("test.dat", "hello");
    return 0;
}

```

- A) hello
- B) new worldhello
- C) new world

D) hello,rld

- 下面程序执行后，输出结果是：_____。

```
#include <stdio.h>
int main()
{
    FILE *fp;
    int i=20, j=30, k, n;
    fp=fopen("d1.dat", "w");
    fprintf(fp, "%d\n", i);
    fprintf(fp, "%d\n", j);
    fclose(fp);
    fp=fopen("d1.dat", "r");
    fscanf(fp, "%d%d", &k, &n);
    printf("%d %d\n", k, n);
    fclose(fp);
    return 0;
}
```

A) 20 30

B) 20 50

C) 30 50

D) 30 20

- 如下程序的输出结果是：_____。

```
#include <stdio.h>
int main()
{
    FILE *fp;
    int i, k=0, n=0;
    fp=fopen("d1.dat", "w");
    for(i=1; i<4; i++) fprintf(fp, "%d", i);
    fclose(fp);
    fp=fopen("d1.dat", "r");
    fscanf(fp, "%d%d", &k, &n);
    printf("%d %d", k, n);
    fclose(fp);
    return 0;
}
```

A) 1 2

B) 123 0

C) 1 23

D) 0 0

- 如下程序的输出结果是：_____。

```
#include <stdio.h>
int main()
{
```

```

FILE *fp;
char str[]="HELLO";
fp=fopen("CON", "w");
fputs(str, fp);
fclose(fp);
return 0;
}

```

- A) 在屏幕上显示"HELLO"
 - B) 把"HELLO"存入 PRN 文件中
 - C) 在打印机上打印出"HELLO"
 - D) 以上都不对
- 以下程序是将一个名为 old.txt 的文件复制到一个名为 new.txt 的新文件中。请在两个空白处选择适当的选项。

```

#include <stdio.h>
int main()
{
    int c;
    FILE *fp1, *fp2;
    if((fp1=fopen("old.txt", __(1) ))==NULL)
    {
        printf("cannot open old file\n");
        exit(0);
    }
    if((fp2=fopen("new.txt", __(2) ))==NULL)
    {
        printf("cannot open new file\n");
        exit(0);
    }
    c=fgetc(fp1);
    while(c!=EOF)
    {
        fputc(c, fp2);
        c=fgetc(fp1);
    }
    fclose(fp1);
    fclose(fp2);
    return 0;
}

```

(1)

- A) "r"
- B) "r+"
- C) "rb"
- D) "rb+"

(2)

- A) “w+”
- B) “wb+”
- C) “w”
- D) “wb”

- 以下程序企图把从终端输入的字符输出到名为 abc.txt 的文件中，直到从终端读入字符 # 时结束输入和输出操作，但程序有错。

```
#include <stdio.h>
int main()
{
    FILE *fout;
    char ch;
    fout=fopen('abc.txt', 'w');
    ch=fgetc(stdin);
    while(ch!='#')
    {
        fputc(ch, fout);
        ch=fgetc(stdin);
    }
    fclose(fout);
    return 0;
}
```

出错的原因是：_____。

- A) 函数 fopen 调用形式错误
- B) 输入文件没有关闭
- C) 函数 fgetc 调用形式错误
- D) 文件指针 stdin 没有定义

2. 填空

- 在 C 语言中，文件的存取是以 (1) 为单位的，这种文件被称为 (2) 文件。
- FILE *p 的作用是定义了一个 (1)，其中的 FILE 是在 (2) 头文件中定义的。
- 在 C 程序中，文件按存取方式分为 (1) 和 (2) 两种类型。
- 在 C 语言的文件系统中，最重要的概念是文件指针，定义文件指针的方法是 (1) *fp;，其中 fp 是一个指向 (2) 类型结构体的指针变量。
- 在 C 语言文件系统中，文件指针指向 (1)，文件内部的位置指针指示文件内部的 (2)。

3. 简答

- 问题：
 - 什么是文件型指针？
 - 通过文件指针访问文件有什么好处？

4. 程序运行结果

- 以下程序的功能是：_____。

```
#include <stdio.h>
int main()
{
    FILE *fp;
    long num=0L;
    if((fp=fopen("abc.txt", "r"))==NULL)
    {
        printf("Open error\n");
        return 0;
    }
    while(!feof(fp))
    {
        fgetc(fp);
        num++;
    }
    printf("num=%ld\n", num-1);
    fclose(fp);
    return 0;
}
```

- 已有文本文件 test.txt，其中的内容是 Hello,everyone!。以下程序中，文件 test.dat 已正确为“读”而打开，由文件指针 fp 指向该文件，则程序的输出是：_____。

```
#include <stdio.h>
int main()
{
    FILE *fp;
    char str[40];
    fp=fopen("test.txt", "r");
    fgets(str, 5, fp);
    printf("%s\n", str);
    fclose(fp);
    return 0;
}
```

5. 程序完形填空

- 若 fp 已正确定义为一个文件指针，data.dat 为二进制文件，请填空，以便为“读”而打开此文件：fp=fopen(_____)。
- 以下程序用来统计文本文件（每行不超过 80 个字符）中的字符行数，请填空。

```
#include <stdio.h>
int main()
```



```

{
    FILE *fp;
    int num=0;
    char buff[80];
    if((fp=fopen("abc.txt", "r"))==NULL)
    {
        printf("Open error\n");
        return 0;
    }
    while(!feof(fp))
    {
        fgets(____);
        num++;
    }
    printf("lines=%d\n", num);
    fclose(fp);
    return 0;
}

```

- 下面程序把从终端读入的文本用@作为文本结束标志，并将该文本写入新文件 a.txt 中，请填空。

```

#include <stdio.h>
FILE *fp;
int main()
{
    char ch;
    if((fp=fopen( (1) ))==NULL) return 0;
    while((ch=getchar())!='@') fputc(ch, fp);
    (2);
    return 0;
}

```

- 以下程序的功能是：从键盘上输入一个字符串，把该字符串中的小写字母转换为大写字母，输出到文件 test.dat 中，然后从该文件读出字符串并显示出来。请填空。

```

#include <stdio.h>
int main()
{
    FILE *fp;
    char str[100];
    int i=0;
    if((fp=fopen("test.dat", (1)))==NULL)
    {
        printf("不能打开该文件\n");
        return 0;
    }
    printf("input a string: ");

```

```
gets(str);
while(str[i])
{
    if(str[i]>='a' && str[i]<='z') str[i]= (2);
    fputc(str[i], fp);
    i++;
}
fclose(fp);
fp=fopen("test.dat", (3));
fgets(str, 100, fp);
printf("%s\n", str);
fclose(fp);
return 0;
}
```