

一、单项选择（30分）

1. 已知大写字母A的ASCII码值是65，小写字母a的ASCII码是97，则用八进制表示的字符常量'\101'是（A）。【3】

- A、字符A
- B、字符a
- C、字符e
- D、非法的常量

2. 下列程序段的输出结果为（C）。【4】

```
float k=0.8567; printf("%4.1f%%",k*100);
```

- A、85.6%
- B、85.6%%
- C、85.7%
- D、.857

3. 下面的scanf函数输入数据,使得a=10, b=15, s1='S', s2='s', x1=1.5, x2=-3.75, x3=67.8, 正确的键盘输入方法是(A)。(用"[CR]"表示回车, U表示空格) 【4】

```
main()
{int a,b;
 float x1,x2,x3;
 char s1,s2;
 scanf("%5d%5d%c%c%f%f*f%f", &a, &b, &s1, &s2, &x1, &x2, &x3);
}
```

- A、10UUU15Ss1.5U-3.75U+1.5U67.8[CR]
- B、10UUU15USUs1.5U-3.75U+1.5U67.8[CR]
- C、10UUU15USUsU1.5U-3.75U+1.5U67.8[CR]
- D、10UUU15Ss1.5U-3.75U+1.5,67.8[CR]

4. C语言的if语句中,用作判断的表达式为(A)。【5】

- A、任意表达式
- B、算术表达式
- C、逻辑表达式
- D、关系表达式

5. 以下程序的输出结果是:(A) 【5】

```
main()
{int m=5;
```

```

    if(m++>5) printf("%d\n",m);
    else printf("%d\n",m--);
}

```

- A、 6
- B、 5
- C、 7
- D、 4

**//: , ? & || 控制表达式 (if switch for while) for 三个表达式**

6. 以下for循环的执行次数是 for(x=0, y=0; (y=123)&&(x<4); x++); (C) 。 【6】

- A、 3次
- B、 循环次数不定
- C、 4次
- D、 是无限循环

7. 以下程序的执行结果是(B) 。 【6】

```

main()
{   int   num = 0;
    while( num <= 2 ) {   num++;   printf( "%d, ", num ); }   }

```

- A、 0, 1, 2
- B、 1, 2, 3,
- C、 1, 2, 3, 4,
- D、 1, 2,

8. 以下不能正确进行字符串赋初值的语句是 (A) 。 【7】

- A、 char str[5]="good!";
- B、 char str[5]={'g', 'o', 'o', 'd'};
- C、 char \*str="good!";
- D、 char str[]="good!";

9. 函数调用: strcat(strcpy(str1, str2), str3)的功能是(D) 。 【7】

- A、 将串str1复制到串str2中后再连接到串str3之后
- B、 将串str2连接到串str1之后再再将串str1复制到串str3中
- C、 将串str1连接到串str2之后再复制到串str3之后
- D、 将串 str2 复制到串 str1 中后再将串 str3 连接到串 str1 之后

10. 以下函数的类型是(C) 。 【8】

```

fff(float x)
{
    return 5;
}

```

- A、 void 类型
- B、 与参数x的类型相同

- C、int类型
- D、无法确定

11. 设有如下函数【8】

```
fac(int x)
{
    float y=1.0;
    return(x*y);
}
```

则函数的类型为(A).

- A、int型
- B、float型
- C、无法确定
- D、void 型

12. 设有宏定义#define PI 3.14 和#define S(r) PI\*r\*r, 则S(2)的值为(B). 【8】

- A、6.28
- B、12.56
- C、3.14
- D、9.42

13. 下列关于指针定义的描述, 错误的是(A)。【9】

- A、指针是一种变量, 该变量用来存放某个变量的值
- B、指针是一种变量, 该变量用来存放某个变量的地址值
- C、指针变量的类型与它所指向的变量类型一致
- D、指针变量的命名规则与标识符相同

14. 以下程序的输出结果是(A)。【9】

```
main()
{char str[ ]="ABCD", *p=str;printf("%d\n", *(p+3));}
```

- A、68
- B、不确定的值
- C、字符D的地址
- D、0

15. 对结构体类型的变量的成员的访问, 无论数据类型如何都可使用的运算符是(B)。【10】

- A、&
- B、.
- C、\*
- D、->

二、判断 (20 分)

- 1. 在C语言中, 各种类型的整型数据在内存中都占2个字节。F【3】
- 2. 若有 int i=10, j=2; 则执行完 i\*=j+8; 后 i 的值为 28。F【3】

3. 格式字符%g 选用%e 或%f 格式中输出宽度较长的一种格式输出实数. F 【4】
4. C 语言程序中输入一个中间带有空格的字符串时, 可以用 scanf 函数. F 【4】
5. 设 d=1, e=2, f=3, 则逻辑表达式!(d+e)+f&&e+f\*2 的值为 0. F 【5】
6. 已知 a=4, b=2, c=3, d=5, 表达式 a>b?a:c<d?c:d 的值为 4. T 【5】
7. 语句 if(a>b) printf("%d", a); else printf("%d", b); 可以用语句 printf("%d", a>b?a:b); 取代. T 【5】
8. 在 do-while 循环中, 任何情况下都不能省略 while. T 【6】
9. break 语句能够终止当前进行的多层循环. F 【6】
10. continue 语句对于 while 和 do-while 循环来说, 意味着转去计算 While 表达式. T 【6】
11. 对静态变量的初始化不是在编译阶段完成的. F 【8】
12. gets 函数是用来输入一个字符的函数. F 【7】
13. 设有数组定义:char array[]="hello"; 则数组 array 所占的内存空间为 5 字节. F 【7】
14. 当全局变量与局部变量同名时, 在局部变量的作用域范围之内, 局部变量起作用. T 【8】
15. C 语言中, 通过函数调用只能获得一个返回值. F 1801 【8】
16. 数组名作为函数调用时的实参, 实际上传递给形参的是数组第一个元素的值. F 【8】
17. 被包含文件与其所在的文件(即用#include 命令的源文件), 在预编译后, 成为同一个文件. T 【8】
18. 如果需要向函数内传递多个值, 用数组和用指针效果是一样的. T 【9】
19. 两个类型相同的结构体变量, 可以互相赋值. T 【10】
20. C 语言中, 在打开文件时, 必须说明文件的使用方式, "wb+"只能用于文件的写. F 【11】

### 三、填空 (16 分)

1. int x; x=-3+4%-5\*3, 则 x 的值为\_\_\_\_. 9 【3】
2. 下列语句输出的结果是\_\_\_\_, unsigned int b=65535;printf("%d", b);-1 【3】
3. x=5, y=8 时, c 语言表达式 5-2>=x-1<=y-2 的值是\_\_\_\_. 1 【5】
4. 设x和y均为int型变量, 则以下for循环中的scanf语句最多可执行的次数是\_\_\_\_. 3 【6】  
for (x=0, y=0; y!=123&& x<3; x++)  
scanf ("%d", &y);
5. 连接字符串的函数是\_\_\_\_, 只写函数名即可. strcat 【7】
6. 从函数的形式上看, 函数分为无参函数和\_\_\_\_两种类型. 有参函数 【8】
7. 定义 int \*p, a[5]; 将数组 a 的首地址赋给指针变量 p 的语句是\_\_\_\_. p=a; p=&a[0]; 【9】
8. 设有以下结构类型说明和变量定义, 则变量 a 在内存所占字节数是 \_\_\_\_。34 struct stud { char name[10]; float s[4]; double ave; } a, \*p; 【10】

### 四、程序设计 (16 分)

1. /\*-----  
【程序设计】  
-----\*/

功能: 编写函数fun(str, i, n), 从字符串str中删除第i个字符开始的连续n个字符 (注意:str[0]代表字符串的第一个字符)。

-----\*/

```

#include "stdio.h"
#include "string.h"
main()
{
    char str[81];
    int i,n;
    void fun(char str[],int i,int n);
    void TestFunc();
    printf("请输入字符串str的值:\n");
    scanf("%s",str);
    printf("你输入的字符串str是:%s\n",str);
    printf("请输入删除位置i和待删字符个数n的值:\n");
    scanf("%d%d",&i,&n);
    while (i+n-1>strlen(str))
    {
        printf("删除位置i和待删字符个数n的值错! 请重新输入i和n的值\n");
        scanf("%d%d",&i,&n);
    }
    fun(str,i,n);
    printf("删除后的字符串str是:%s\n",str);
    TestFunc();
}

void fun(char str[],int i,int n)
{
    /*****Begin*****/

while(str[i+n-1])
    {
        str[i-1]=str[i+n-1];
        i++;
    }
    str[i-1]='\0';

    /***** End *****/
}

void TestFunc()
{
    FILE *IN,*OUT;
    char i[200];
    /*char o[200];*/
    IN=fopen("in.dat","r");

```

```

    if(IN==NULL)
    {
        printf("Read File Error");
    }
    OUT=fopen("out.dat","w");
    if(OUT==NULL)
    {
        printf("Write File Error");
    }
    fscanf(IN,"%s",i);
    fun(i,2,8);
    fprintf(OUT,"%s\n",i);

    fclose(IN);
    fclose(OUT);
}

```

2. /\*-----

### 【程序设计】

功能：请编一个函数void fun(int tt[M][N],int pp[N])，  
 tt指向一个M行N列的二维数组，求出二维数组每列  
 中最小元素，并依次放入pp所指一维数组中。二维  
 数组中的数已在主函数中赋予。

-----\*/

```

#include "conio.h"
#include "stdio.h"
#define M 3
#define N 4

void fun(int tt[M][N],int pp[N])
{
    /*****Begin*****/
    int i,j;
    for(j=0;j<N;j++)
    {
        pp[j]=tt[0][j];
        for(i=1;i<M;i++)
            if(tt[i][j]<pp[j])
                pp[j]=tt[i][j];
    }
}

```

```

    }

    /***** End *****/
}

main()
{
    int t[M][N]={ {22, 45, 56, 30}, {19, 33, 45, 38}, {20, 22, 66, 40} };
    void NONO();
    int p[N], i, j, k;
    printf("The original data is:\n");
    for(i=0; i<M; i++)
    {
        for(j=0; j<N; j++)
            printf("%6d", t[i][j]);
        printf("\n");
    }
    fun(t, p);
    printf("\nThe result is:\n");
    for(k=0; k<N; k++) printf("%4d", p[k]);
    printf("\n");
    NONO();
}

void NONO()
{
    int i, j, array[3][4], p[4];
    FILE *rf, *wf ;
    rf = fopen("in.dat", "r") ;
    wf = fopen("out.dat", "w") ;
    for (i=0; i < 3; i++)
        for (j=0; j < 4; j++)
            fscanf(rf, "%d", &array[i][j]);
    fun(array, p);
    for (j=0; j < 4; j++)
    {
        fprintf(wf, "%7d", p[j]);
        fprintf(wf, "\n");
    }
    fclose(rf) ;
    fclose(wf) ;
}

```

## 五、程序填空（10 分）

1. /\*-----

### 【程序改错】

功能：求1到20的阶乘的和。

```
-----*/  
#include "stdio.h"  
void fun()  
{  
    int n, j;  
    float s=0.0, t=1.0;  
    for(n=1; n<=20; n++)  
    {  
        /******ERROR*****/  
        s=1;    // t=1;  
        for(j=1; j<=n; j++)  
            /******ERROR*****/  
            t=t*n;    // t=t*j;  
        /******ERROR*****/  
        s+t=s;    // s=s+t;    s+=t;  
    }  
    /******ERROR*****/  
    printf("jiecheng=%d\n", s);    //printf("jiecheng=%f\n", s);  
}  
  
main()  
{  
    fun();  
}
```

2. /\*-----

### 【程序改错】

功能：编写一个函数，该函数可以统计一个长度为3的字符串在另一个字符串中出现的次数。

例如：假定输入的字符串为：asdassdfgasdaszx67asdmklo，字符串为：asd, 则应输出n=4。



```

-----*/

#include "stdio.h"
#include "string.h"
#include "conio.h"
int fun(char *str, char *substr)
{
    /******ERROR******/
    int i, n=0    // int i, n=0;
    /******ERROR******/
    for(i=0; i<=strlen(str); i++) // for(i=0; i<=strlen(str)-3; i++)
                                //for(i=0; i<strlen(str)-2; i++)
        if((str[i]==substr[0])&&(str[i+1]==substr[1])&&(str[i+2]==substr[2]))
            /******ERROR******/
            ++i;    // n++; n+=1; n=n+1; ++n;

    return n;
}

main()
{
    char str[81], substr[4];
    int n;

    printf("输入主字符串: ");
    gets(str);
    printf("输入子字符串: ");
    gets(substr);
    puts(str);
    puts(substr);
    n=fun(str, substr);
    printf("n=%d\n", n);
}

```

## 六、程序改错（8分）

1. /\*-----\*/

### 【程序填空】

功能：计算平均成绩并统计90分以上人数。

-----\*/

```

#include "stdio.h"
main()
{
    int n,m;float grade,average;
    /*****FILL*****/
    average=n=m=____; //0
    while(1)
    {
        /*****FILL*****/
        ____("%f",&grade); //scanf
        if(grade<0) break;
        n++;
        average+=grade;
        /*****FILL*****/
        if(grade<90) ____; //continue
        m++;
    }
    if(n) printf("%.2f%d\n",average/n,m);
}

```

2. /\*-----  
**【程序填空】**  
 -----\*/

功能：求出二维数组中的最大元素值。

```

-----*/
#include "stdio.h"
/*****FILL*****/
int max_value(int m,int n,int ____ ) // array[][4] array[3][4]
{
    int i,j,max;
    max=array[0][0];
    for(i=0;i<m;i++)
        for(j=0;j<n;j++)
            /*****FILL*****/
            if(max<array[i][j]) ____; // max=array[i][j]
    return(max);
}
main()
{
    int a[3][4]={ {1,3,5,7}, {2,4,6,8}, {15,17,34,12} };
    /*****FILL*****/

```

```
printf("max value is %d\n", ____); //max_value(3, 4, a)
}
```