

## 一、选择题

1 定义结构体数组：

```
struct stu
{
    int num;
    char name[20];
} X[5] = {1, "LI", 2, "ZHAO", 3, "WANG", 4, "ZHANG", 5, "LIU"};
for(i=1; i<5; i++)
    printf("%d%c", x[i].num, x[i].name[2]);
```

以上程序段的输出结果为( )。

A. 2A3N4A5UB. 112A3H4IC. 1A2N3A4UD. 2H3A4H5I

参考答案：A

参考解析：本题主要考查结构体数组的使用。x[i].num 为结构体 x1~x5 中的 num 成员，x[i].name[2] 是结构体 x[i] 中 name 成员的第 3 个元素。程序执行循环过程时：第 1 次循环，i=1，输出 x[1].num，x[1].name[2] 的值；第 2 次循环，i=2，输出 x[2].num，x[2].name[2] 的值，即 3N；第 3 次循环，i=3，输出 x[3].num，x[3].name[2] 的值，即 4A；第 4 次循环，i=4，输出 x[4].num，x[4].name[2] 的值，即 5U。

2 下述程序的输出结果是( )。

```
#define N 10
void fun(int a[], int n, int m)
{
    int i, j;
    for(i=m; i>n; i--) a[i+1]=a[i];
}
main()
{
    int i, a[N]={1,2,3,4,4,5,6,7,8,9,10};
    fun(a, 2, 9);
    for(i=0; i<5; i++) printf("%d", a[i]);
}
```

A. 10234B. 12344C. 12334 D. 12234

参考答案：B

参考解析：函数 fun() 的功能是从数组 a 中依次将当前值赋给一个元素，所以当函数 fun(a, 2, 9) 执行后，数组 a[] = {1, 2, 3, 4, 4, 5, 6, 7, 8, 9, 10}，由题意依次取该数组前 5 个元素，所以选项 B 正确。

3 数据的存储结构是指( )。

- A. 数据所占的存储空间
- B. 数据的逻辑结构在计算机中的存放形式
- C. 数据在计算机中的顺序存储方式
- D. 存储在计算机外存中的数据

参考答案：B

参考解析：数据的存储结构，又称为数据的物理结构，是数据的逻辑结构在计算机中的存放形式。

4 下面描述中，不属于软件危机表现的是( )。

- A. 软件过程不规范
- B. 软件开发生产率低
- C. 软件质量难以控制
- D. 软件成本不断提高

参考答案：A

参考解析：软件危机的表现包括：①对软件开发的进度和费用估计不准确；②用户对已完成的软件系统不满意的现象时常发生；③软件产品的质量往往靠不住；④软件常常是不可靠的；⑤软件通常没有适当的文档；⑥软件成本在计算机系统总成本中所占的比例逐年上升；⑦软件开发生产率提高的速度远远跟不上计算机应用迅速普及深入的趋势。

5 下列运算符中优先级最高的算符是( )。

- A. !

B. %

C. -=

D. &&

参考答案: A

参考解析: C 语言中规定运算符的优先顺序: 一元算术运算符→二元算术运算符→关系运算符→逻辑运算符→赋值运算符, A 项是关系运算符, B 项和 D 项都是逻辑运算符, C 项是赋值运算符, 因此优先级最高的是 A 项。

6 利用 fseek() 函数可实现的操作是( )。

A. 实现文件的顺序读写

B. 改变文件的位置指针

C. 实现文件的随机读写

D. 以上答案均正确

参考答案: D

参考解析: 本题考查 fseek() 函数, 其功能是: 改变文件的位置指针; 辅助实现文件的顺序读写; 辅助实现文件的随机读写。

7 有以下程序:

```
#include <stdio.h>
main()
{ char ch;
  ch='A'+5-'3';
  printf("%d,%c\n",ch,ch);
}
```

程序的输出结果是( )。

A. 67. CB. B, CC. C, D D. 不确定的值

参考答案: A

参考解析: `char ch;ch='A'+5-'3';`//计算字符 `'A'+5-'3'` 即 `'A'+2`, 因为 `'5'` 与 `'3'` ASCII 码就差 2, 得 `'C'` `printf("%d,%c\n",ch,ch);`//%d 是按整型输出, 输出的是 ch 的 ASCII 码, 查表可得 67//%c 是按字符型输出, 输出为 C 所以输出就为 67, C。因此 A 为正确选项。

8 以下叙述中正确的是( )。

- A. do-while 语句构成的循环不能用其他语句构成的循环来代替
- B. do-while 语句构成的循环只能用 break 语句退出
- C. 用 do-while 语句构成循环时, 只有在 while 后的表达式为非零时结束循环
- D. 用 do-while 语句构成循环时, 只有在 while 后的表达式为零时结束循环

参考答案: D

参考解析: 选项 A, 可以用 for 循环代替; 选项 B, 不满足条件时即条件的逻辑值为假(0)时退出, break 也可以; 选项 C, while(条件), 条件为真时进行循环, 为假(0)时退出。

9 读取二进制文件的函数调用形式为: `fread(buffer, size, count, fp);`, 其中 buffer 代表的是( )。

- A. 一个文件指针, 指向待读取的文件
- B. 一个整型变量, 代表待读取的数据的字节数
- C. 一个内存块的首地址, 代表读入数据存放的地址
- D. 一个内存块的字节数

参考答案: C

参考解析: fread 函数中的参数 buffer 是内存块的首地址, 输入的数据存入此内存块中。

10 若变量 a、b、x、j 都已正确定义并赋值, 则以下符合 C 语言语法的表达式是( )。

- A. `a+&=(b-4)*(a=3)`
- B. `x%(-3)`
- C. `a=a*3=2D`
- D. `y=float(i)`

参考答案: A

参考解析: 取余运算符%两边都应是正数, 赋值语句左边必须是变量, 不能是表达式, 强制转换语句强制类型应加括号, 而非加在变量上。

11 下列合法的声明语句是( )。

A. int \_abc=50; B. double int=3+5e2.5; C. long do=1L; D. float 3\_asd=3e-3;

参考答案: A

参考解析: 标识符是由字母、数字或下划线组成, 并且它的第一个字符必须是字母或者下划线。B 选项 int 不是表达变量类型的标识符, 它不能再用作变量名和函数名。C 选项 do 是 C 语言的一个关键字。D 选项标识符不符合只能以字母或下划线开始的要求。

12 下列程序的输出结果是( )。

```
int f1(int x,int y){return x>y? x:y;}
int f2(int x,int y){return x>y? y:x;}
main()
{ int a=4,b=3,c=5,d=2,e,f,g;
  e=f2(f1(a,b),f1(c,d));
  f=f1(f2(a,b),f2(c,d));
  g=a+b+c+d-e-f;
  printf("%d,%d,%d\n",e,f,g);}
```

A. 4, 3, 7 B. 3, 4, 7 C. 5, 2, 7 D. 2, 5, 7

参考答案: A

参考解析: 根据函数的定义可知, 如果  $x > y$  成立, 则函数  $f1$  返回  $x$  的值, 函数  $f2$  返回  $y$  的值。所以  $f1(a, b) = 4$ ,  $f1(c, d) = 5$ ,  $f2(a, b) = 3$ ,  $f2(12, d) = 2$ , 故  $e = f2(4, 5) = 4$ ,  $f = f1(3, 2) = 3$ ,  $g = 4 + 3 + 5 + 2 - 4 - 3 = 7$ 。

13 下列程序的运行结果是( )。

```
#include<stdio.h>
#define ADD(x) x+x
main()
{ int m=1,n=2,k=3;
  int s=ADD(m+n)*k;
  printf("s=%d",s);
}
```

A. sum=18 B. sum=10 C. sum=9 D. sum=25

参考答案: B

参考解析: 本题考查带参数的宏的定义: “#define ADD(x)x+x”中 x 代表 m+n, 故 s 写成  $s=m+n+m+n*k=1+2+1+2*3=10$ 。

14 下面程序段的输出结果是( )。

```
int i=32770; printf("%d\n", i);
```

A. 32769 B. 3211 67 C. -32766 D. 输出不确定的数

参考答案: C

参考解析: 在 Turbo C 2.0 环境中, int 数据类型在内存中只占两个字节, 其表示形式为二进制补码形式, 所以范围是:  $-32768 \sim 32767$ , 即  $100000000000000002 \sim 011111111111111111$  (首位为符号位)。而 32770 转换为二进制为  $10000000000000102$ , 其符号位为 1, 故输出时为一个负数, 而它的绝对值是求负后的结果, 补码求负的方法是: 先按位求反, 再加一。所以求得绝对值为  $01111111111111111012+1=01111111111111102$ , 即十进制的 32766。综上所述, 选项 C 正确。

15 执行下述程序后, 输出的结果是( )。

```
#include<stdio.h>
#define S(X) X*X
```

```
void main()
{ int a=9,k=3,m=2;
  a/=S(k+m)/S(k+m);
  printf("%d",a);
}
```

A. 1 B. 4 C. 9 D. 0

参考答案: D

参考解析: 本题主要是考查带参数的宏定义, 过程如下:  $\text{int } a = a / S(k+m) / s(k+m) = a / (k+m * k+m / k+m) = 9 / (3+2 * 3+2 / 3+2 * 3+2) = 0$ 。

16 若希望下列的程序运行后输出 25, 程序空白处的正确选项是( )。

```
main()
{  int i,j=50,a[]={7,4,10,5,8};
    for (    )
        j+=a[i];
    printf("%d",j-40);
}
```

A.

$i=1; i < 4; ++i$

B.

$i=1; i < 3; ++i$

C.

$i=4; i > 2; i--$

D.

$i=2; i < 4; ++i$

参考答案: D

参考解析：要想使程序输出是 25，则在 printf 中，j-40=25，j=65，而 j 初值是 50，“j+=a[i];”也可写成 i=j+a[i];，j=50+a[i]-65，当 a[i]=15，条件满足. 数组下标由 0 开始，当 i=2 时，指向元素 10，i=3 时，10+5=15，退出循环体，所以选项 D 正确。

17 运行下列程序，若从键盘输入字母“a”，则输出结果是( )。

```
char c;  
c=getchar();  
if(c>='a'&&.<='g')c=c+4;  
else if(c>='g'&&.<='z')c=c-21;  
else printf("input error! \n");  
putchar(c);
```

A. FB. tC. e D. d

参考答案：C

参考解析：getchar 函数是读取键盘的输入函数，本题中输入字母“a”，则 c='a'，判断 if 条件，发现 'a'>='a'&&.<='g' 成立，则执行下列语句：c=c+4，c 得到的字符的 ASCII 码值，退出 if 语句，通过 putchar 函数输出字符。

18 若 fp 是指向某文件的指针，且尚未读到文件末尾，则函数 feof(fp) 的返回值是( )。

A. EOF-1 B. 非零值 D. 0

参考答案：D

参考解析：本题考查 feof(fp) 函数，其功能是：测试所指的文件的位置指针是否已达到文件尾，如果已达到文件尾，则函数返回非 0 值；否则返回 0，表示文件未结束。

19 有以下程序



```
#include <stdio.h>
main()
{ int y=9;
  for (;y>0;y--)
    if (y%3==0) printf ("%d",--y);
}
```

程序的运行结果是( )。

A. 741B. 963C. 852 D. 875421

参考答案: C

参考解析: 在 for 循环中, 当 y 的值为 9、6 或 3 时, 执行输出语句, 输出表达式“--y”的值。y 先自减 1, 然后再输出 y 的值。

20 若希望下列的程序运行后输出 25, 程序空白处的正确选项是( )。

```
main()
{ int i=50, a[]={7,4,10,5,8};
  for ( )
    j+=a[i];
  printf("%d",j-40);
}
```

A.

```
i=1;i<4;i++
```

B.

```
i=1;i<3;i++
```

C.

```
i=4;i>2;i--
```

D.

$i=2; i<4; ++i$

参考答案: D

参考解析: 要想使程序输出 25, 则  $j-40=25$ ,  $j=65$ , 而  $j$  初值是 50, 所以填入 for 循环的语句, 使引用的数组元素累加  $65-50=15$  即可。D 选项中,  $i$  下标从 0 开始,  $i=2$  指向第 3 个数  $50+a[2]=50+10=60$ , 继续循环  $60+a[3]=60+5=65$ , 此时  $i=4$  跳出, 循环结束。

21 有以下程序:

```
#include <stdio.h>
struct tt
{int x; struct tt *y;} *a;
struct tt a[4] = {20, 30, 15, 17};
main()
{int i;
 struct tt *p=a;
 for(i=1; i<=2; i++) {printf("%d, ", p->x); p=p->y;}
```

程序的运行结果是( )

A. 20, 30, B. 30, 17, C. 15, 30, D. 20, 15,

参考答案: D

参考解析: 程序的结构体数组  $a$  的元素形成了一个单向的循环链表, 每个元素的指针成员都指向下一个元素的地址。

22 有以下程序:

```
#include <stdio.h>
main()
{ int y=10;
  for(;y>0;y--)if(y%3==0)
  { printf("%d",--y);continue;
  }
}
```

程序的输出结果是( )。

A. 741B. 852C. 963 D. 875421

参考答案: B

参考解析: 进行循环, 若 y 能被 3 整除, 则输出--y, 如:

y=9, 满足 y%3==0, 输出--y, 也就是 8

y=8, 不满足 y%3==0

y=7, 不满足 y%3==0

y=6, 满足 y%3==0, 输出--y, 也就是 5

依次类推……

结果为 852, 因此正确选项为 B。

23 若程序中有宏定义行: #define N 100, 则以下叙述中正确的是( )。

A. 宏定义行中定义了标识符 N 的值为整数 100

B. 在编译程序对 C 源程序进行预处理时用 100 替换标识符 N

C. 对 C 源程序进行编译时用 100 替换标识符 N

D. 在运行时用 100 替换标识符 N

参考答案: B

参考解析: C 语言中, 凡是以“#”号开头的行, 都称为“编译预处理”。对 C 源程序进行编译前, 由编译预处理程序对这些编译预处理命令进行处理。

24 下列程序的输出结果是( )。

```
void fun(int *x,int y)
{ printf("%d%d", *x, *y); *x=3; *y=4;}
main()
{ int x=1,y=2;
  fun(&y,&x);
  printf("%d%d",x,y);
}
```

A. 2 1 4 3B. 1 2 1 2C. 1 2 3 4D. 2 1 1 2

参考答案: C

参考解析: 当执行 fun 函数时, 因为参数传递时是地址传递, 因此形参值的改变会影响实参中的数, 在执行 fun 函数时先输出 1 2, 接着把 \*x 赋 3, \*y 赋 4, 此时带回主函数中, 输出的值为 3 4。

25 在执行完下列的 C 语言代码之后, 则 8 的值是( )。

```
char a='A';
int b;
B=((348&56)&&(&a<'b'));
```

A. 0B. 1C. TRUE D. FALSE

参考答案: B

参考解析: 本题考查 3 个知识点: 将一个字符赋给一个字符变量时, 是将该字符对应的 ASCII 码存储到内存单元中; 常见 ASCII 码的值如 A 和 a; 只要逻辑与运算符 && 的两个运算对象都为真时, 返回值就是 1。

26 若变量已正确定义为 float 型, 要通过输入语句:

```
scanf("%f %f%f", &a, &b, &c); 给 a 赋值 11.0, b 赋值 22.0, c 赋值 33.0, 不正确的输入形式是( )。
```

A. 11 22 33 B. 11.0, 22.0, 33.0 C. 11.0 D. 1122 22.0 33.0

参考答案: B

参考解析: B 选项中不应该加 “,” 号。

27 有以下程序:

```
#include <stdio.h>
main()
{ int w=4,x=3,y=2,z=1;
  printf("%d\n", (w<x? w:z<y? z:x));
}
```

程序的输出结果是( )。

A. 1 B. 2 C. 3 D. 4

参考答案: A

参考解析: 条件运算符具有右结合性。当一个表达式中出现多个条件运算符时, 应该将位于最右边的问号与离它最近的冒号配对, 并按这一原则正确区分各条件运算符的运算对象。w

28 有以下程序:

```
#include <stdio.h>
void fun(float *a, float *b)
{ float w;
  *a = *a + *a; w = *a; *a = *b; *b = w;
}
main()
{ float x=2.0, y=3.0, *px=&x, *py=&y;
  fun(px, py); printf("%2.0f, %2.0f\n", x, y);
}
```

程序的输出结果是( )。

A. 4, 3 B. 2, 3 C. 3, 4 D. 3, 2

参考答案: C

参考解析:

【解析】#include <stdio.h>

```
void fun(float *a, float *b) // a 即指向 px 的指针, b 即指向 py 的指针, 则 *a 指向 x, *b 指向 y
{
    float w;
    *a = *a + *a; // 将 x 地址存放的数据自加, x 变为 4
    w = *a;
    *a = *b;
    *b = w; // 将 x 和 y 地址存放的数据互换, 则 x 变为 3, y 变为 4
}

main()
{
    float x = 1.0, y = 3.0, *px = &x, *py = &y; // px 和 py 分别是指向 x 和 y 地址的指针
    fun(px, py);
    printf("%2.0f, %2.0f\n", *px, *py); // %2.0f 表示结果前面有 2 个空格, 小数点后有 0 位, 即保留到个位
}
```

29 若变量 c 为 char 类型, 能正确判断出 c 为小写字母的表达式是( )。

A.

'a' <= c <= 'z'

B.

(c >= 'a') || (c <= 'z')

C.

$(\text{'a'} \leq c) \text{and} (\text{'z'} > c)$

D.

$(c > \text{'a'}) \& \& (c < \text{'z'})$

参考答案: D

参考解析: 字符型数据在计算机内部是以 ASCII 存储的, 英文大写字母和小写字母在 ASCII 码表中都是连续的, 大写字母 A 到 Z 是从 65 到 90 的, 小写字母 a 到 z 是从 97 到 122, 所以只要变量 c 大于 a 并且小于 z 就能保证其为小写字母。

30 以下程序的输出结果是\*

```
main()
{
    int x=0.5; char z='a';
    printf("%d\n", (x&1)&&(z<'z'));
```

A. 0 B. 1 C. 2 D. 3

参考答案: B

参考解析:  $(x \& 1) \& \& (z < \text{'z'}) = (0.5 \& 1) \& \& (\text{'a'} < \text{'z'}) = 1 \& \& 1 = 1$ , 故选择 B 选项。

31

在 16 位编译系统上, 若有定义 `int a[] = {10, 20,`

30), \*p = &a;, 当执行 `"p++;"` 后, 下列说法错误的

是( )。

- A. p 向高地址移了一个字节
- B. p 向高地址移了一个存储单元
- C. p 向高地址移了两个字节
- D. p 与 a+1 等价

参考答案: A

参考解析: 因为 p 是整型的指针变量, 因此其移动一个位置即两个字节, 故选择 A 选项。

32 若有以下程序:

```
#include <stdio.h>
void sub(double *y, double *z) { *Y = *
y-1.0; *z = *z+x; }
main()
{ double a=2.5, b=9.0, *pa, *pb;
  pa=&a; pb=&b;
  sub(b-a, pa, pa); printf("%f\n", a);
}
```

程序的输出结果是( )。

- A. 9.000000 B. 1.500000 C. 8.000000 D. 10.500000

参考答案: C

参考解析: 从函数参数来看,  $x=b-a=6.5$ ;  $y=z$  是指向 a 的指针 pa, 也就是说  $*y=*z=2.5$ ; 由于指针作为参数, 可以将函数中计算得到的对指针引用的  $*y, *x$  作为最后计算得到的 a 的值, 从函数语句看, 类似做这样的运算,  $a=a-1.0; a=a+x$ ; 这里  $x=6.5$  于是计算得到,  $a=8$ ; 由于输出格式为浮点型, 输出结果为 8.000000。因此正确答案为 C。

33 概要设计中要完成的任务是( )。

- A. 系统结构和数据结构的设计



B. 系统结构和过程的设计

C. 过程和接口的设计

D. 数据结构和过程的设计

参考答案: A

参考解析: 在 c 语言中软件概要设计的基本任务是设计软件系统结构、数据结构及数据库设计、编写概要设计文档和概要设计文档评审。

34 下述程序的输出结果是( )。

```
#include<stdio.h>
void main()
{ int b[5]={2,4,6,8,10,12};
  int *p=b; **q=p;
  printf("%d", *(p++));
  printf("%d", **q);
}
```

A. 4, 4B. 2, 2C. 2, 4D. 2,

参考答案: C

参考解析: 首先定义了一个指向一维数组的指针 p, 一个指向指针 p 的指针变量 q; 输出\*(p++)是去输出\*p, 即 b[0] 的值, 再将指针 p 指向数组的下一个元素 b[1]; 输出\*\*q 是输出指针 p 所指单元的内容。

35 设有以下定义:

```
union data
{int d1;float d2;}demo;
```

则下面叙述中错误的是( )。

A. 变量 demo 与成员 d2 所占的内存字节数相同

B. 变量 demo 中各成员的地址相同

C. 变量 demo 和各成员的地址相同

D. 若给 demo.D1 赋 99 后, demo.d2 中的值是 99.0

参考答案: D

参考解析: 共用体变量中的所有成员共享一段公共存储区, 所有成员的首地址相同。在每一个时刻所有的成员中只有一个有效, 即只有最后一个被赋值的成员有效, 其余的成员无效。

36 下列程序的运行结果是( )。

```
int y=5,x=14;  
y=((x=3*y,x+1),x-1);  
printf("x=%d,y=%d",x,y);
```

A. x=27, y=27 B. x=12, y=13 C. x=15, y=14 D. x=y=27

参考答案: C

参考解析: 逗号表达式的解题思路是从左到右, 依次求解, 整个逗号表达式的值就是最后一个表达式的值。表达式  $(x=3*y, x+1)$  中, 依次是  $x=3*y=3*5=15$ , 表达式  $x+1=16$ , 但没有给  $x$  或  $y$  赋值, 所以  $x$  还是等于 15; 第 3 个表达式  $x-1=15-1=14$ , 所以  $y$  的值为 14。

37 合法的 C 语言赋值语句是( )。

A. a=b=58 B. k=int(a-4-b); C. a=58, b=58 D. --1;

参考答案: B

参考解析: 赋值语句两个标志: 赋值(就是“=”)和语句(加上“;”)。选项 A, 缺少分号, 应该改为  $a=b=58$ ; 选项 B, 错误, 类型转换不是这么写, 应该是这样:  $k=(int)(a+b)$ ; C 选项中缺少分号; 选项 D, 相当于  $i=i-1$ , 所以也可以算作赋值操作。

38 下列程序的输出结果是( )。

```

#include<stdio.h>
#include<string.h>
void fun(int b[])
{ static int i=0;
  do
  { b[i]=b[i+1];
    }while(++i<2);
}
main()
{ int k,a[5]={1,3,5,4,9};
  fun(a);
  for(k=0;k<5;k++) printf("%d",a[k]);
}

```

A. 13579B. 48579C. 48549 D. 48999

参考答案: C

参考解析: 本题考查 do-while 循环。在 fun 函数中, 首先定义了静态变量 i=0, do-while 循环要实现的功能是以 b[i]=b[i+1], b[1]=b[1]+b[2], 其他元素不变的规则重新给 b 数组赋值。在主函数中, 通过调用 fun 函数, 按上面的规则对数组 a 重新赋值, 最后输出数组 a。

39 若有定义语句: int a[2][3], \*p[3];, 则以下语句中正确的是( )。

A. p=a;

B. p[0]=a;

C. p[0]=&a[0][2];

D. p[1]=&a;

参考答案: B

参考解析: 选项 B 表示把整型数组 a[0] 的指针赋给 p[0]。

40 设在 C 语言中, float 类型数据占 4 个字节, 则 double 类型数据占( )个字节。

A. 1B. 2C. 8 D. 4

参考答案：C

参考解析：C语言中各种数据类型在内存中所占的字节数与机器的位数有关，16位机中(也就是在标准C中)，若int类型数据占2个字节，则float类型数据占4个字节，double类型数据占8个字节。

## 二、程序填空题

41 请补充函数proc()，该函数的功能是：把字符串str中的字符按字符的ASCII码升序排列，处理后的字符串仍然保存在原串中，字符串及其长度作为函数参数传入。

例如，如果输入“9fedcba”，则输出为“abcdefg”。

注意：部分源程序给出如下。

请勿改动main()函数和其他函数中的任何内容，仅在函数proc()的横线上填入所编写的若干表达式或语句。

试题程序：

```
#include <stdio.h>
#include <stdlib.h>
#define N 80
void proc(char str[], int n)
{
    int i, j;
    char ch;
    for(i=0; i<n; i++)
        for(j=___【1】___; j<n; j++)
            if(str[i]>str[j])
            {
                ch=str[j];
                ___【2】___;
                str[i]=ch;
            }
}
void main()
{
    int i=0, strlen=0;
    char str[N];
```

```

system("CLS");
printf("\nInput a string:\n");
gets(str);
while(str[i]!='\0')
{
    strlen++;
    i++;
}
proc(str, strlen);
printf("\n * * * display string * * *\n");
puts(str);
}

```

参考解析：

【1】 i+1 【2】 str[j]=str[i]

【解析】：题目要求把字符串 str 中的字符按其 ASCII 码值升序排列，可以使字符串中的每一个字符与其后的所有字符相比较，因此【1】填“i+1”；把该字符与 ASCII 码值小于其本身而且位置在它后面的字符相交换，最后就可以使 ASCII 码值小的在前、大的在后，因此【2】填“str[j]=str[i]”。

### 三、程序改错题

42 下列给定程序中，函数 proc() 的功能是：删除字符串 str 中所有空白字符(包括 Tab 字符、Enter 符及换行符)。输入字符串时以“#”结束输入。

请修改程序中的错误，使它能得出正确的结果。

注意：不要改动 main() 函数，不得增行或删行，也不得更改程序的结构。

试题程序：

```

#include <string.h>
#include <stdio.h>
#include <ctype.h>
void proc(char *p)
{
    int i, t; char c[80];
    for(i=0,t=0;p[i];i++)
        if(! isspace(* (p+i))) c[t++] = p[i];
    // * * * * found * * * *
    c[t] = "\0";
    strcpy(p,c);
}

void main()
{
    char c, str[80];
    int i=0;
    printf("Input a string: ");
    c = getchar();
    while(c != '#')
    {
        str[i] = c; i++; c = getchar();
    }
    str[i] = '\0';
    proc(str);
    puts(str);
}

```

参考解析:

错误: `c[t] = "\0";`

正确: `c[t] = '\0';`

【解析】删除空格后的字符串放在数组 c 中, 完成字符的复制后, 要为新的字符串添加结束符。因此, “`c[t] = "\0";`”应改为 “`c[t] = '\0';`”。

#### 四、程序设计题

43 请编写一个函数 `unsigned proc(unsigned u)`, `u` 是一个大于 10 的无符号整数, 若 `u` 是 `n` ( $n \geq 1, n \geq 2$ ) 位的整数, 则函数求出 `u` 的后 `n-1` 位的数作为函数值返回。

例如, `u` 值为 1234, 则函数返回 234; 若 `u` 值为 234, 则函数返回 34。

注意: 部分源程序给出如下。

请勿改动 main() 函数和其他函数中的任何内容，仅在函数 proc() 的花括号中填入所编写的若干语句。

试题程序：

```
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
unsigned proc(unsigned u)
{
}

void main()
{
    unsigned x;
    system("CLS");
    printf("Enter an unsigned integer number: ");
    scanf ("%u", &x);
    printf("The original data is: %u\n", x);
    if(x<0)
        printf("Data error! ");
    else
        printf ("The result: %u\n", proc(x));
}
```

参考解答：

```
unsigned proc(unsigned u)
{
    if(u>=10000) //此时 u 为五位，则返回其后的四位
        return u%10000;
    if(u>=1000) //此时 u 为四位，则返回其后的三位
        return u%1000;
    if(u>=100) //此时 u 为三位，则返回其后的两位
        return u%100;
    if(u>=10) //此时 u 为两位，则返回其后的一位
        return u%10;
}
```

【解析】要返回无符号整数的后  $n-1$  位，首先要判断无符号整数的位数  $i$  然后通过求余来实现。

全才学习网  
www.100xuexi.com