

## 一、选择题

1 关系表中的每一行记录称为一个( )。

A. 字段 B. 元组 C. 属性 D. 关键码

参考答案: B

参考解析: 在关系表中, 每一列称为一个属性, 对应表中的一个字段; 每一行称为一个元组, 对应表中的一条记录。

2

C 语言规定, 函数返回值的类型是( )。

- A) 由调用该函数时的主调函数类型所决定
- B) 由 return 语句中的表达式类型所决定
- C) 由调用该函数时系统临时决定
- D) 由定义该函数时所指定的数值类型决定

参考答案: D

参考解析: 本题考查函数调用时的数值类型。函数调用时, 函数返回值的类型既不由主调函数类型所决定, 也不由 return 语句中的表达式类型所决定, 更不是由系统临时决定, 而是由定义该函数时所指定的数值类型所决定。

3 在一棵二叉树上, 第 5 层的结点数最多是( )。

A. 8 B. 9 C. 15 D. 16

参考答案: D

参考解析: 根据二叉树的性质: 二叉树第  $i$  ( $i > 1$ ) 层上至多有  $2^{i-1}$  个结点。得到第 5 层的结点数最多是 16 个。

4

有以下程序

```
#include <stdio.h>
int fun(int a,int b)
{ if(b==0) return a;
  else return(fun(--a,--b));
}
main()
```

```
{ printf("%d\n", fun(4,2));}
程序的运行结果是( )。
```

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

参考答案: B

参考解析: 由程序可知, 函数 fun(int a, int b) 是一个递归函数。所以当主函数中调用 fun(4, 2) 时, 其执行过程如下: fun(4, 2) → fun(3, 1) → fun(2, 0), 其返回值为 2。所以正确答案为选项 B)。

5

下面的 for 语句的循环次数为 ( )。

```
for(y=1; y=0; (y!=19)&&(x<6));
```

- A) 是无限循环
- B) 循环次数不定
- C) 最多执行 6 次
- D) 最多执行 5 次

§ 考试大 在线 考试中心 §

参考答案: D

参考解析: 本题考查 for 循环。只考虑 x 的取值变化, x 从 1 取到 5, 可以循环 5 次, 但是并不知道 y 是如何变化的, 有可能出现 y=19 提前跳出循环的情况, 所以是最多执行了 5 次。

6

下述程序的输出结果是 ( )。

```
#include<stdio.h>
void main()
{ int a[20],*p[4];
  int i, k=0;
  for(i=0;i<20;i++)
    a[i]=i;
  for(i=0;i<4;i++)
    p[i]=&a[i*i+1];
  for(i=0;i<4;i++)
  { k=k+*p[i];}
  printf ("%d",k);
}
```

A. 10 B. 18 C. 6

参考答案: B

参考解析: 第一个 for 循环, 给数组 a 的 20 个元素分别赋值 0~19; 第二个 for 循环, 给数组 P 赋值,  $p[0]=a[1]=1$ ,  $p[1]=a[2]=2$ ,  $p[2]=a[5]=5$ ,  $p[3]=a[10]=10$ ; 第三个 for 循环, 将数组 p 中元素的值进行累加,  $k=1+2+5+10=18$ 。

7

下述程序的运行结果是 ( )。

```
for(i=5;i<7;i++) printf((i%2)? ("%d\n"): ("##%d\n"),i);
```

A) \*\*2  
##4  
\*\*5  
\*\*6

B) ##3  
\*\*4  
##5  
\*\*6

C) ##3  
\*\*4  
##5  
##6

D) \*\*3  
##4  
\*\*5  
##6

来源: 考试大在线考试中心  
ks.233.com

参考答案: D

参考解析：本题考查 printf 函数的输出格式和条件表达式“?:”的使用。printf 函数输出时“\*\*%d\n”和“##%d\n”中的“#”和“\*\*”都是直接输出。②条件表达式的基本格式为：“a?b:c”。它的运算规则为：当 a 为非 0 时，整个表达式的值取 b 的值，当 a 为 0 时，整个表达式取 c 的值。

8 运行下面程序时，从键盘输入字母 A，则输出结果是( )。

```
#include  
  
main() {char ch;  
  
ch=getchar();  
  
switch(ch)  
  
{case 'A': printf("AAAA\n");  
case 'B': printf("BBBB\n");  
default: printf("CCCC\n");  
}  
}
```

A. AAAA

B. AAAA

C. BBBB

C. AAAA

BBBB

CCCC

D. AAAA

CCCC

参考答案: C

参考解析: 本题考查 switch 语句。因为 ch=A, 所以第一个 switch 语句, 执行 case~A 后面的语句, 输出 AAA A. 没有遇到 break 语句, 所以接着再执行 case' B' 和 default, 分别输出 BBBB 和 CCCC。

9 若  $x=2$ ,  $y=3$ , 则  $x\&y$  的结果是( )。

A. 0B. 2 C. 3D. 5

参考答案: B

参考解析: 本题主要考查按位与 (&) 运算的知识点。“&”是对两个运算量相应的位进行逻辑与, “&”的规则与逻辑与“&&”相同。 $x=2$  的二进制为 00000010,  $y=3$  的二进制为 00000011,  $x\&y=00000010$ , 转化为十进制即为 2。

10

有以下程序:

```
#include <stdio.h>
int fun(int *k)
{
    int b=0;
    b=*k+b;
    *k=fun(b);
    return b;
}
main()
{
    int a[10]={1,2,3,4,5,6,7,8};
    for(i=2;i<4;i++){ printf("%d", fun(&a[i])); }
    printf("\n");
}
```

程序运行后的输出结果是( )。

A. 10 12B. 8 10C. 3 10D. 10 28

参考答案:

第 5 页: 填空题

11 下列说法正确的是

A. `int (**def) [10];`----def 指向一个指针, 该指针指向 10 个整型变量构成的数组

- B. `int>(*gh[10]);`---gh 指向一个指针, 该指针指向 10 个整型变量构成的数组
- C. `int(*gh[10]);`---gh 是一个指针, 指向由 10 个指向整型变量的指针构成的数组
- D. A, C 均正确

参考答案: D

参考解析: `int(*gh[10]);`---gh 是一个指针, 指向由 10 个指向整型变量的指针构成的数组, `int(**def)[10];`---def 指向一个指针, 该指针指向 10 个整型变量构成的数组。

12 算法的有穷性是指( )

- A. 算法程序的运行时间是有限的
- B. 算法程序所处理的数据量是有限的
- C. 算法程序的长度是有限的
- D. 算法只能被有限的用户使用

参考答案: A

参考解析: 算法的基本特征主要包括以下四个方面:

- 可行性: 针对实际问题而设计的算法, 执行后能够得到满意的结果。
- 确定性: 算法中的每一个步骤都必须有明确的定义, 不允许有模棱两可的解释和多义性。
- 有穷性: 算法必需在有限时间内做完, 即算法必需能在执行有限个步骤之后终止。
- 拥有足够的情报: 要使算法有效必需为算法提供足够的情报当算法拥有足够的情报时, 此算法才最有效的; 而当提供的情报不够时, 算法可能无效。

13 以下数据结构中, 属于非线性数据结构的是( )。

- A. 栈 B. 线性表 C. 队列 D. 二叉树

参考答案: D

参考解析：二叉树属于非线性结构。栈是一种特殊的线性表，这种线性表只能在固定的一端进行插入和删除操作；队列可看作是插入在一端进行，删除在另一端进行的线性表。

14 在下列叙述中，错误的是( )。

- A. C 语言中，二维数组或多维数组是按行存放的
- B. 赋值表达式  $b[1][2]=a[2][3]$  是正确的
- C.  $\text{char } a[1];a[0]='A'$  与  $\text{int } a[1];a[0]='A'$  等价
- D. 数组名后的方括号内可以为常量表达式，也可以为变量

参考答案：D

参考解析：本题考查数组的 4 个知识点：①C 语言中，二维数组或多维数组元素排列的顺序是按行存放；②不同数组的元素之间可以相互赋值；③字符型数组中存放的是字符，int 型数组中存放的是字符对应的 ASCII 码值，虽然在计算机内部字符常量都是作为整型量来处理的，其对应的整数值就是 ASCII 码值，但两者并不完全等价；④数组名后的方括号内可以为常量，也可以为常量表达式，但不能为变量。

15 在下述程序中，判断  $i>j$  共执行的次数是

```
main()
{
    int i=0, j=10, k=2, s=0;
    for (;;)
    {
        i+=k;

        if (i>j)
        {
            printf("%d", s);

            break;

        }
        s+=i;
    }
}
```

}

A. 4

B. 7

C. 5

D. 6

参考答案: D

参考解析: 分析程序,  $i+=k$ ; 使得  $i$  的值每次增加 2 然后再和  $j$  进行比较看  $i>j$  是否成立,  $i$  的变化过程为:  $i=2, i=4, i=6, i=8, i=10, i=12$ , 此时  $i>j$  成立, 所以判断  $i>j$  共执行的次数为 6 次。

16

下面程序运行后的输出结果是 ( )。

```
#include<stdio.h>
main()
{
    int a[5]={1,4,2,0}, b[5]={1,2,3,4,5}, i, s=0;
    for(i=0; i<5; i++) s=s+a[i];
    printf("%d\n", s);
}

for(i=6; i<10; i++)
for(j=0; j<4; j++){...}
```

A. 12 B. 15 C. 11 D. 8

参考答案: A

参考解析: 这道题目是计算机一维数组  $b$  中某些元素的和。这些元素的下标值是一维数组  $a$  中元素值。  $s=0+b[1]+b[0]+b[4]+b[2]+b[0]=2+1+5+3+1=12$ 。

17 在关系代数运算中, 有 5 种基本运算, 它们是 ( )。

A. 并( $\cup$ )、差( $-$ )、交( $\cap$ )、除( $\div$ )和笛卡儿积( $\times$ )



- B. 并( $\cup$ )、差( $-$ )、交( $\cap$ )、投影( $\pi$ )和选择( $\sigma$ )
- C. 并( $\cup$ )、交( $\cap$ )、投影( $\pi$ )、选择( $\sigma$ )和笛卡儿积( $\times$ )
- D. 并( $\cup$ )、差( $-$ )、投影( $\pi$ )、选择( $\sigma$ )和笛卡儿积( $\times$ )

参考答案: D

参考解析: 并、差、笛卡儿积、投影和选择是 5 种基本的运算, 其他运算即交、连接和除, 均可以通过 5 种基本的运算来表达。

18 数据库系统在其内部具有 3 级模式, 用来描述数据库中全体数据的全局逻辑结构和特性的是( )。

- A) 外模式  
B) 概念模式  
C) 内模式  
D) 存储模式
- 来源: 考友太在线考试中心  
15.2.3.6m

参考答案: B

参考解析: 概念模式, 也称逻辑模式, 是对数据库系统中全局数据逻辑结构的描述, 是全体用户应用公共数据视图。它不涉及具体的硬件环境与平台, 与具体的软件环境也无关。

19 有以下字符串, 说明表达式 `strlen(s)` 值的是( )。

```
char s[10]={'a', '\n', 'a', 'b', '\t', 'c'};
```

- A. 10 B. 1 C. 6 D. 7

参考答案: C

参考解析: `strlen(s)` 返回 `s` 字符串中字符的个数, 若 `S` 是一个数组, 则返回数组元素的个数。

20 以下选项中正确的语句组是( )。

A. char s[];s="HELLO!";B. char\*s;s={"HELLO!"};C. char s[10];s="HELLO!";D. char \*s;s="HELLO!";参考答案: D 我的答案: 未作答收起解析收藏纠错

参考解析: 本题考查数组和字符串的赋值。选项 A)

中数组 s 没有明确其大小就赋值是错误的。选项 B) 中不能有大括号, 选项 C) 中的写法应该为 {'H','E','L','L','O','!'}。

第 5 页: 填空题

21 以下不正确的定义语句是( )。

A) double x[5]={1.0, 0.0, 0.4, 0.5, 0.0};  
B) int y[5]={0, 1, 2, 3, 4, 5};  
C) char s1[]={'1', '2', '3', '4', '5'};  
D) char c2[]={'a', 'b', 'c'};

参考答案: B

参考解析: 本题考查对一维数组的赋值。选项 B) 中, y[5] 说明数组中有 5 个元素, 但却给它赋了 6 个值, 数组溢出所以是错误的。

82 内聚性是对模块功能强度的衡量, 下列选项中, 内聚性较弱的是( )。

A. 顺序内聚 B. 偶然内聚 C. 时间内聚 D. 逻辑内聚

参考答案: B

参考解析: 内聚是从功能角度来衡量模块的联系, 它描述的是模块内的功能联系。内聚有如下种类, 它们之间的内聚度由弱到强排列:

偶然内聚——模块中的代码无法定义其不同功能的调用。但它使该模块能执行不同的功能, 这种模块为巧合强度模块。

逻辑内聚——这种模块把几种相关的功能组合在一起, 每次被调用时, 由传送给模块的参数来确定该模块应完成哪一种功能。

时间内聚——这种模块顺序完成一类相关功能，比如初始化模块。它顺序地为变量置初值。

过程内聚——如果一个模块内的处理元素是相关的，而且必须以特定次序执行，则称为过程内聚。

通信内聚——这种模块除了具有过程内聚的特点外，还有另外一种关系，即它的所有功能都通过使用公用数据而发生关系。

顺序内聚——如果一个模块内各个处理元素和同一个功能密切相关，而且这些处理必须顺序执行，处理元素的输出数据作为下一个处理元素的输入数据，则称为顺序内聚。

功能内聚——如果一个模块包括为完成某一具体任务所必需的所有成分，或者说模块中所有成分结合起来是为了完成一个具体的任务，此模块则为功能内聚模块。

23

以下程序段的输出结果是（ ）。

```
#include<stdio.h>
main()
{ char p[3][4]={"ABC", "DEF", "GHI"};
  int i;
  for(i=0;i<3;i++)
    puts(p[i]);
}
```

- A) A  
B  
C
- B) A D G  
A B C  
D E F  
G H I
- C) A B C  
D E F  
G H I
- D) A B C

参考答案：C

参考解析：本题首先定义了一个字符串数组P，通过for循环输出p[0]、p[1]、p[2]的值。

定义如下变量和数组：

```
int i, x[3][3]={1,2,3,4,5,6,7,8,9};
则下面语句的输出结果是 ( )。
for(i=0;i<3;i++) printf("%d",x[i][2-i]);
```

- A) 159
- B) 147
- C) 357
- D) 369

参考答案：C

参考解析：本题主要考查二维数组的使用。题中利用  $n$  个  $\text{printf}$  循环来输出  $x$  数组中的指定元素：①当  $i=0$  时，输出  $x[0][2]=3$ ；②当  $i=1$  时，输出  $x[1][1]=5$ ；③当  $i=2$  时，输出  $x[2][0]=7$ 。

设有如下关系表

R			S			T		
A	B	C	A	B	C	A	B	C
4	5	6	4	5	6	4	5	6
5	6	4	10	9	4			
7	8	9						

则下列操作正确的是 ( )。

- A)  $T=R/S$
- B)  $T=R \times S$
- C)  $T=R \cap S$
- D)  $T=R \cup S$

参考答案：C

参考解析：T 就是关系 R 中有的，同时 S 中也有的有序组 (4、5、6)，即交运算 ( $\cap$ )。

下面 4 个关于 C 语言的结论中错误的是 ( )。

- A) 可以用 do...while 语句实现的循环一定可以用 while 语句实现
- B) 可以用 for 语句实现的循环一定可以用 while 语句实现
- C) 可以用 while 语句实现的循环一定可以用 for 语句实现
- D) do...while 语句与 while 语句的区别仅是关键字 “while” 出现的位置不同

参考答案: D

参考解析: 本题考查几个循环语句的关系。①do...while 语句、while 语句、for 语句所实现的循环可以相互替代;②do...while 语句和 while 语句的区别是 do...while 语句至少执行一次, 再判断循环条件, while 语句是先判断条件再执行。

27 下面程序的输出结果是 ( )

```
#include<stdio.h>
#include<string.h>
main()
{ char p1[]="WHO",p2[]="who",str[50]="xyz";
  strcpy(str+1, strcat(p1,p2));
  printf("%s", str); }
```

- A. A) xyzWH0who B) zWH0who C) yWH0who D) xWH0who

参考答案: D

参考解析: 程序中, strcat(p1, p2) 实现将以 p2 为首地址的字符串连接到以 p1 为首地址的字符串后面, 并取消原来 p1 字符串后面的串结束标志, 得到 WH0who; strcpy(str+1, strcat(p1, p2)) 实现将 WH0who 拷贝到以 str+1 开头的地址单元中, 最后得到 xWH0who。

已知一个文件中存放若干工人档案记录，其数据结构如下：

```
struct a
{ char number [100];
  int age;
  float p[6];};
```

定义一个数组：struct a number[10];

假定文件已正确打开，不能正确地从文件中读入10名工人数据到数组b中的是( )。

- A) fread(b,sizeof(struct a),10,fp);
- B) for(i=0;i<10;i++)  
fread(b[i],sizeof(struct a),1,fp);
- C) for(i=0;i<10;i++)  
fread(b+i,sizeof(struct a),1,fp);
- D) for(i=0;i<5;i+=2)  
fread(b+i,sizeof(struct a),2,fp);

参考答案：B

参考解析：数据块输入/输出函数的调用格式为：fread(buffer, size, count, fp)。

fread 函数参数说明：“buffer”是一个指针，对 fread 来说，它是读入数据的存放地址。对 fwrite 来说，是要输出数据的地址。“size”是要读写的字节数：“count”是要进行读写多少个 size 字节的数据项；“fp”是指文件型指针。选项 B) 中 b[i] 是一个数组，不是指针，需改为 &b[i]。

注意：完成一次读操作(fread())后，如果没有关闭流(fclose())，则指针(FILE\*fp)自动向后移动前一次读写的长度，不关闭流则继续下一次读操作，接着上次的输出继续输出。

29. 以下程序的输出结果是( )。

```
#include<stdio.h>
struct st
{ int x;int *y;} *p;
int dt[4]={1,2,3,4};
struct st aa[4]={2,&dt[0],3,&dt[0],4,&dt[0],5,&dt[0],};
main()
{ p=aa;
  printf("%d\n",++(p->x));
}
```

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

参考答案: C

参考解析: 程序首先将指针 P 指向结构体数组 aa 的首地址, 通过 p->x 引用了结构体数组 aa[0] 的第一个成员 2; ++(p->x) 是将 p->x 的值加 1, 所以输出结果为 3。

30 下面结构体的定义语句中, 错误的是( )。

```
A) struct ord {int x; int y; int z;}; struct ord a;  
B) struct ord {int x; int y; int z;} ;ord a;  
C) struct ord {int x; int y; int z;}a;  
D) struct {int x; int y; int z;} a;
```

参考答案: B

参考解析: 在选项 B) 中, 即使在定义结构体 ord 后, 不能把 ord 看成是一种数据类型。

第 5 页: 填空题

31 下列说法正确的是( )。

A. main 函数必须放在 C 程序的最后面

B. main 函数必须放在 C 程序的最前面

C. main 函数可以放在 C 程序的中间部分, 但在执行 C 程序时是从程序开头执行的

D. main 函数可以放在 C 程序的中间部分, 但在执行 C 程序时是从 main 函数开始的

参考答案: D

参考解析: 每个 C 程序有且只有一个主函数(main), 且程序必须从 main() 函数开始执行, 别外 main() 函数可以放在程序中的任意位置。

32 有以下程序, 其中%u 表示按无符号整数输出( )。

```
main()
```

```
{unsigned int x=0xFFFF; /*x 的初值为十六进制数*/
```

```
printf("%u\n", x);
```

```
)
```

程序运行后的输出结果是( )。

A. . 1B. 65535C. 32767D. 0xFFFF

参考答案: B

参考解析:

因为%u表示按无符号整数输出,而十六进制表示为 0xFFFF,所以输出无符号整数的最大值 65535。

33 以下关于 C 语言的叙述中正确的是( )。

A. 预处理命令通常位于函数体外面,但也可以位于函数体中间

B. C 语言中的变量定义必须在其他语句之前

C. 在 C 语言程序的书写中,一个语句必须独占一行

D. 一个 C 语言源程序可以由一个或多个源文件组成,每个源文件可由自己的 main 函数

参考答案: A

参考解析:

源程序可以有预处理命令(include 命令仅为其中的一种)。C 语言中的变量可以在使用之前的任何位置进行定义。C 语言程序的书写中,多个语句可以位于一行,但为了增强可读性,一般每行只有一条语句。一个 C 语言源程序可以由一个或多个源文件组成,但仅可以有一个 main 函数。



有以下程序段：

```
int i,n;
for(i=0;i<8;i++)
{
    n=rand()%5;
    switch(n)
    {
        case 1:
        case 3:printf("%d\n",n);break;
        case 2:

        case 4:printf("%d\n",n);continue;
        case 0:exit(0);
    }
    printf("%d\n",n);
}
```

以下关于程序段执行情况的叙述，正确的是（ ）

- A) for 循环语句固定执行 8 次
- B) 当产生的随机数  $n$  为 4 时结束循环操作
- C) 当产生的随机数  $n$  为 1 和 3 时不做任何操作
- D) 当产生的随机数  $n$  为 0 时结束程序运行

参考答案：D

参考解析：

当产生的随机数  $n$  为 0 时，执行 `case 0: exit(0);` 这条语句，结束程序的运行。

35 若变量已正确定义，有以下程序段：

```
i=0;
do printf("%d",i);while(i++);
printf("%d\n",i);
其输出结果是（ ）。
```

- A. 0    B. 1    C. 1, 1D. 程序进入无限循环

参考答案：B

参考解析：do...while 循环的执行过程如下：①执行 do 后面的循环体中的语句。②计算 while 后一对圆括号中表达式的值。当值为非 0 时，转去执行步骤①；当值为 0 时，执行步骤③。③退出 do...while 循环。对于本题变量  $i$  的初始值等于“0”，接着执行

do 后面的循环体中的语句，输出变量 i 的值 0。再判断 while 后面括号中表达式 i++ 的值，其值为 0，所以循环结束。此时变量 i 的值经过自加已经变为 1，所以再次输出 i 的值“1”。

36 数字字符 0 的 ASCII 值为 48，运行下列程序的输出结果是( )。

```
main()
{
    char a='1',b='2';
    printf("%c", b++);
    printf("%d\n", b-a);
}
```

A. 3, 2B. 50, 2C. 2, 2D. 2, 50

参考答案: C

37 当输入为“Fool?Swallow”时，下面程序的执行结果是( )。

```
#include <stdio.h>
main()
{
    char c;
    while(c!='?')
    {
        c=getchar();
        putchar(c);
    }
}
```

A. SwallowB. FoolC. Fool?D. ?Swallow

参考答案: C

参考解析：本程序是通过 getchar() 函数读入字符，并通过 putchar() 函数将字符逐个输出，当用户输入“?”时停止输出。

38 设 x 和 y 都是 int 类型，且 x=1, y=2，则 printf(“%d%d”, x, y, (x, y)) 的输出结果是( )。

A. 12B. 122C. 1, 2D. 输出值不确 参考答案: A

参考解析:

本题考查 printf 函数的格式和逗号表达式。printf 函数在输出时, 是根据格式说明符的个数输出的, 如果函数中有多余的项将不予输出: 如果有输出项少于格式说明符, 则出错, 不能输出正确的结果。

本题中逗号表达式 (x, y) 只返回 Y 的值。此时格式说明的个数是 2, 而输出项的个数是 3, 因此 printf 函数只输出前两项。

39 下列叙述中错误的是( )。

A. 用户所定义的标识符允许使用关键字 B

. 用户所定义的标识符应尽量做到“见名知意”

C. 用户所定义的标识符必须以字母或下划线开头

D. 用户定义的标识符中大、小写字母代表不同标识

参考答案: A

参考解析: 在 C 语言中, 合法标识符的命名规则是: 标识符可以由字母、数字和下划线组成, 并且第一个字符必须为字母或下划线, 凡是要求标识符的地方都必须按此规则命名。在 C 语言的标识符中, 大写字母和小写字母被认为是两个不同的字符。在选择标识符时, 应注意做到“见名知义”, 即选择具有一定含义的英文单词或汉语拼音作为标识符, 以增加程序的可读性。

40 若变量都已正确说明, 则以下程序段输出结果为( )

。

```
#include<stdio.h>
main()
{char a='a';
int b=2;
printf(a>b?"***a=%d\n"###b=%d",a,b);
}
```

A) \*\*\*a=3

B) ###b=3

C) \*\*\*a=3##b=5

D) 全部错误

参考答案: D

参考解析: 本题考查 printf 函数的格式和三元运算符“?:”的使用。printf 函数中格式说明符之前插入的任何字符都原样输出。本题中 a>b 为真, 所以返回“\*\*\*a=97”。

## 二、填空题

41

下面程序的运行结果是\_\_\_\_\_。

```
#define P(a) printf("%d ",a)
main()
{ int j[]={1,2,3,4,5,6,7},i=5;
  for(j=3;j>1;j--)
  { switch(j)
    { case 1:
      case 2: P(a[i++]); break;
      case 3: P(a[--i]);
    }
  }
}
```

参考解析: 5 5

【解析】分析程序执行过程，第一次循环时，j=3，i=5，因为 switch(3)，所以执行 case 3，调用 P(a[--i])=p(a[4])=P(5)，输出 5；第二次循环时，j=2，i=4，因为 switch(2)，所以执行 case 2，调用 P(a[i++])=P(8[4])=P(5)，输出 5，之后 i 自加 1 等于 5。

42 下面程序的输出结果是\_\_\_\_\_。

```
main()
{ int i=0,a=0;
  while(i<20)
  { for(;;)
    { if((i%10)==0)break;
      else i--;}
    i+=11;
    a+=i;
  }
  printf("%d\n",a);
}
```

参考解析：32

【解析】while(表达式)的功能是：首先计算表达式的值，若为真，则执行循环体语句，执行完毕，再计算表达式的值，若仍为真，则重复执行循环体语句，直到表达式的值为假，结束 while 语句的执行，继续执行 while 语句后面的语句；i=0 时满足循环条件，也满足 if 的条件，执行语句得 i=11，a=11，第 2 次循环不满足 if 的条件，执行 i=i-10，执行 i+=11；a+=7 后，得 i=81，a=22，不满足 i<20，循环结束。

43

定义 int a=5,b=20; 执行语句 printf("%d\n",++a\*--b/5%13); 后，输出的结果为\_\_\_\_\_。

参考解析：9

【解析】此题考查的是算术运算符。依照运算级的优先性，首先执行“++”和“==”，“++a”执行之后 a 的值变为 6，++a\*--b= (++a)+(--b)=6\*19=114；此后再依次执行/和%，++a\*--b/5!= 114/5!=22!=9。

44 在面向对象方法中，\_\_\_\_\_描述的是具有相似属性与操作的一组对象。

参考解析：类

【解析】在面向对象方法中，把具有相同属性和方法的对象称为类，类是对象的抽象，对象是类的实例。

45 在算法正确的前提下，评价一个算法的两个标准是空间复杂度和\_\_\_\_\_。

参考解析：时间复杂度

【解析】在算法正确的前提下，评价一个算法的两个标准是空间复杂度和时间复杂度。算法的时间复杂度，是指执行算法所需要的计算工作量。算法的空间复杂度是指执行这个算法所需要的内存空间。

46 在树形结构中，没有前件的结点是\_\_\_\_\_。

参考解析：

根结点

【解析】树型结构具有如下特点

- 每个结点只有一个前件，称为父结点，没有前件的结点只有一个，称为树的根结点，简称为树的根；
- 每一个结点可以有多个后件，它们都称为该结点的子结点。没有后件的结点称为叶子结点；
- 一个结点所拥有的后件个数称为树的结点度；
- 树的最大层次称为树的深度。

47 符合结构化原则的三种基本控制结构是顺序结构、\_\_\_\_\_和循环结构。

参考解析：选择结构

【解析】程序设计语言仅仅使用顺序、选择和重复(循环)三种基本控制结构就足以表达出各种其他形式结构的程序设计方法。遵循程序结构化的设计原则，按结构化程序设计方法设计出的程序易于理解、使用和维护；可以提高编程工作的效率，降低软件的开发成本。

下面程序的运行结果是\_\_\_\_\_。

```
#include<stdio.h>
#define SIZE 12
main()
{ char s[SIZE];int i;
  for(i=0;i<SIZE;i++) s[i]='A'+i+32;
  sub(s,5,SIZE-1);
  for(i=0;i<SIZE;i++) printf("%c",s[i]);
  printf("\n");
}
sub(char *a,int t1,int t2)
{ char ch;
  while(t1<t2)
  { ch=*(a+t1);
    *(a+t1)=*(a+t2);
    *(a+t2)=ch;
    t1++;t2--;
  }
}
```

参考解析:

abcdetkjilgf

【解析】本题先给字符型数组 S 的 12 个元素赋值 a 到 l 的 12 个字母，函数 sub(char\*a, int t1, int t2) 的功能是将数组 a 的从第 t1+1 个元素到 t2+1 个元素进行逆置，在主函数中调用 sub(s, 5, SIZE-1) 函数，是将数组 S 的第 6 个元素到第 12 个元素进行逆置，其他元素位置不变。

“printf(“%d\n”, strlen(“t\023\xAB\nC”));” 语句的输出结果是\_\_\_\_\_。

参考解析: 6

【解析】strlen()函数的作用是计算字符串的长度并作为函数的返回值，这里的长度不包括串尾的结束标志'\0'。

'\t'是转义字符，代表横向跳若干格；'\\"'是转义字符，代表双引号；'\023'也只代表一个字符，而不管转义字符后面有几个字符；'\xAB'是以两位十六进制数AB表示的ASCII码字符，只代表一个字符；'\n'是转义字符，代表回车换行。

题中语句中的字符串有5个字符常量各代表一个字符，再加上字母C，所以返回的长度是6。

50

以下程序运行后的输出结果是\_\_\_\_\_。

```
#define Power2(x) 2*x  
main()  
{ int i=6,j=8;  
  printf("%d",Power2(i+j));  
}
```

参考解析：

62

【解析】本题考查的是宏定义。带参数的宏定义是按#define命令中指定的字符串从左到右进行转换。本题中替换的式为：6+8\*6+8，代入i，j的值得到62。

51

以下程序的输出结果是\_\_\_\_\_。

```
#include <stdio.h>  
void swap(int *a,int *b)  
{ int *t;  
  t=a;a=b;b=t;  
}  
main()  
{ int i=3,j=5,*p=&i,*q=&j;  
  swap(p,q);printf("%d %d\n",*p,*q);  
}
```



参考解析：3 5

【解析】本题考查函数中形参和实参的传递。在 C 语言函数中实参和形参传递具有不可逆性，参数只能由实参传向形参，而不能由形参传向实参，虽然 swap 函数的功能是实现两个数的交换，但由于没有返回值，故最终的输出结果为 3 5。

52 以下程序的输出结果是\_\_\_\_\_。

```
#include<stdio.h>
main()
{ int x=27;
  while(x>20&& x<30)
  { x++;
    if(x/3){x++;break;}
    else
      continue;
  }
  printf("%d\n",x);
}
```

参考解析：29

【解析】本题考查 while 循环和++运算，x=27 符合循环条件，执行循环体中 x 十十后，x=28，if 条件也成立，执行 x 十十后退出，x 的值增 2，结果 x=29。

53 在结构化分析使用的数据流图中，使\_\_\_\_\_解释其中的图形元素。

参考解析：

数据字典

【解析】数据字典的作用是对数据流图中出现的被命名的图形元素进行确切解释。通常数据字典包含的信息有名称、别名、何处使用/如何使用、内容描述、补充信息等。

54 以下程序的输出结果是\_\_\_\_\_。

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

```

main()
{ int x=0;
  sub(&x,16,2);
  printf("%d\n",x);
}
sub(int*a,int n,int k)
{ if(k<=n) sub(a,n/2,2*k);
  *a+=k;
}

```

参考解析:

14

【解析】本题考查递归函数的调用。在主函数中调用 sub(&x, 16, 2) 函数, 此时 n=16, k=2, if 语句的控制的条件 k=n 成立, 所以执行下面的语句, 调用 sub(x, 8, 4), 此时 n=8, k=4, if 语句的控制条件 k<=n 成立, 所以执行下面的语句, 调用 sub(x, 4, 8), 此时 n=4, k=8, if 语句控制的条件 k<=n 不成立, 因此, 最后得 x=2+4+8=14。

55

下面程序的运行结果是\_\_\_\_\_

```

#include<stdio.h>
main()
{ int a,s,n,m;
  a=2;s=0;n=1;m=1;
  while(m<=4) n=n*a; s=s+n; ++m; }
printf("s=%d\n",s);
}

```

参考解析:

s=30

【解析】分析循环条件“m=1, m<=4”, 所以循环 4 次。第 1 次: n=1\*a=2, s=s+n=2; m=2。第 2 次: n=n\*a=4, s=s+n=6; m=3。第 3 次: n=n\*a=8, s=s+n=14; m=4。第 4 次: n=n\*a=16, s=s+n=30; m=5。循环结束, 输出 s=30。