

一 选择题和填空题

结构体

1. 设有程序段

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

```
int main()
{
    struct date
    {
        int year, month, day;
    }today;
    printf("%d\n", sizeof(struct date));
    return 0;
}
```

设该程序运行结果是 ()。

- A. ~~12~~ B. 10 C. 8 D. 6
2. C 语言结构体类型变量在程序执行期间 ()。
- A. ~~所有成员~~一直驻留在内存中 B. 只有一个成员驻留在内存中
- C. 部分成员驻留在内存中 D. 以上说法都不对
3. 设有语句

```
strcut stu
{
    int a;
    float b;
}stutype;
```

设有以上说明语句，则下面的叙述正确的是 ()。

- A. ~~struct~~ 是结构体类型的关键字
- B. ~~struct stu~~ 是用户定义的结构体类型名
- C. stutype 是用户定义的结构体类型名
- D. ~~a 和 b~~ 都是结构体成员名
4. 设有定义
- ```
struct str
{
 int x;
 float y;
 char z;
}st;
```
- 若有以上程序片段，则下列的叙述中不正确的是 ( )。
- A. struct 是结构体类型的关键字      B. ~~st~~ 是结构体类型名
- C. x, y, z 都是结构体成员名      D. struct str 是结构体类型名
5. 以下结构体类型说明和变量定义中，正确的是 ( )。

A.

```
struct SS
{
 char flag;
 float x;
}
struct SS a,b;
```

没分号

B.

```
struct SS
{
 char flag;
 float x;
};
struct SS a,b;
```

C.

```
struct
{
 char flag;
 float x;
}SS;
SS a,b;
```

D.

```
typedef
{
 char flag;
 float x;
}SS;
SS a,b;
```

6. 若有如下定义，则结构体变量 **b** 占用内存的字节数是 ( )。

```
struct data
{
 int i;
 char ch;
 double f;
}b;
```

A. 1      B. 2      C. 11      D. 16

7. 下列说法错误的是 ( )。

~~A.~~ 设在 visual C++6 环境下运行，则下面结构体定义时，占据了 9 个字节的空间

```
struct student
{
 int num;
 int age;
```

```
char sex;
```

```
}
```

- B. 结构体的成员名可以与程序中的变量名相同
- C. 对结构体中的成员可以单独使用，它的作用相当于普通变量
- D. 结构体可以嵌套定义

8. 设有定义

```
struct sk
```

```
{
```

```
int a;
```

```
float b;
```

```
}data, *p;
```

若有以上程序片段，若有 `p = &data`，则对 `data` 的成员 `a` 的正确引用是（ ）。

- A. `(*p).data.a`
- ☒ B. `(*p).a`
- C. `p->data.a`
- D. `p.data.a`

9. 设定义下列结构体，结构体变量 `P` 的出生年份赋值正确的语句是（ ）。

```
struct st
```

```
{
```

```
int x;
```

```
int y;
```

```
int z;
```

```
};
```

```
struct worker
```

```
{
```

```
char name[20];
```

```
char sex;
```

```
struct st birth;
```

```
}p;
```

- A. `x = 1987`
- B. `birth.x = 1987`
- ☒ C. `p.birth.x = 1987`
- D. `p.x = 1987;`

10. 设有定义

```
struct stu
```

```
{
```

```
int num;
```

```
char sex;
```

```
int age;
```

```
}a1, a2;
```

结构体变量不能整体操作，只能操作成员

若有以上结构体定义，则下列语句中错误的是（ ）。

- ☒ A. `printf("%d, %c, %d", a1);`
- B. `a2.age = a1.age;`
- C. `a1.age++;`
- D. `a1.num = 5;`

11. 说明一个结构体变量时，系统分配给它的内存是（ ）。

- A. 各成员所需内存量的总和
- B. 结构体中第一个成员所需内存量
- C. 成员中占内存量最大者所需的容量
- ☒ D. 综合考虑编译环境和内存对齐的方法及各成员的内存数

12. 若有以下说明和语句，则下面表达式中值为 1002 的是（ ）。

```

struct student
{
 int age;
 int num;
};
struct student stu[3] = {{1001, 20}, {1002, 19}, {1003, 21}};
struct student *p;
p = stu;

```

- A. (p++) -> num      B. (p++) -> age      C. (\*p).num      ~~D. (\*++p).age~~

13. 若有如下定义:

```

struct { int id; char name[10]; } per, *s = &per;

```

则以下对结构体成员的引用中错误的是 ( )。

- A. per.name      B. s->name[0]      ~~C. (\*per).name[8]~~      D. (\*s).id

14. 已知定义

```

struct REC
{
 int num;
 char flag;
 char adr[20];
} rec[10], *pr = rec;

```

设有如上定义, 下面各输入语句中错误的是 ( )。

- ~~A. scanf("%s", &rec.adr);~~      B. scanf("%d", &(\*pr).num);  
 C. scanf("%c", &(pr->flag));      D. scanf("%c", &rec[1].flag);

15. 设有程序段

```

struct student {
 int age;
 int num;
};
struct student stu[3] = {{1001, 20}, {1002, 19}, {1003, 21}};
int main()
{
 struct student *p;
 p = stu;

}

```

若有以上定义和语句, 则以下不正确的引用是 ( )。

- A. (p++) -> num      B. p++  
 C. (\*p).num      ~~D. p = &stu.age~~

16. 设有程序段

```

struct student
{
 int age;
 int num;
} std, *p;

```

p = &std;

若有以下说明和语句，则以下对结构体变量 std 中成员 age 的引用方式正确的是 ( )。

A. std.age

~~B. p->age~~

~~C. (\*p).age~~

D. \*p.age

17. 设有定义

```
struct pupil
```

```
{
```

```
char name[20];
```

```
int sex;
```

```
}pup, *p = &pup;
```

若有以下说明和语句，则对 pup 中 sex 域的正确引用方式是 ( )。

A. p.pup.sex

B. p->pup.sex

C. (\*p).pup.sex

~~D. (\*p).sex~~

18. 设有程序段

```
int main()
```

```
{
```

```
struct cmplx
```

```
{
```

```
int x;
```

```
int y;
```

```
}cnum[2] = {1, 3, 2, 7};
```

```
printf("%d\n", cnum[0].y/cnum[0].x*cnum[1].x);
```

```
return 0;
```

```
}
```

上面程序输出的结果是 ( )。

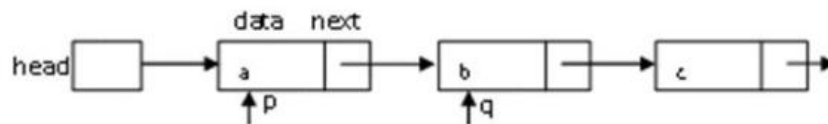
A. 0

B. 1

C. 3

~~D. 6~~

19. 假定已经建立如下链表结构，且指针 p 和 q 已指向如图所示的结点：



则下列选项中可以将 q 所指结点从链表中删除并释放该结点的语句组是 ( )。

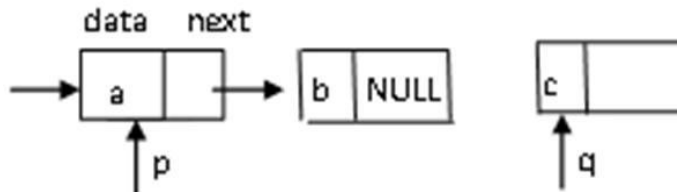
A. (\*p).next = (\*q).next; free(p);

B. p = q->next; free(q);

C. p = q; free(q);

~~D. p->next = q->next; free(q);~~

20. 如果已经建立下面的链表结构，指针 p, q 分别指向如图的结点，则不能将 q 所指向的结点插入到链表末尾的一组语句是 ( )。



A. q->next = NULL; p = p->next; p->next = q;

B. p = p->next; q->next = p->next; p->next = q;

~~C. p = p->next; q->next = p; p->next = q;~~

D. `p = (*p).next; (*q).next = (*p).next; (*p).next = q;`

21. 设有程序段

```
int *p;
p = [] malloc(sizeof(int));
```

若要利用上面程序片段使指针变量 `p` 指向一个存储整型变量的存储单元, 则 `[]` 中应该填入的内容是 ( )。

- A. `int`      B. `int *`      C. `(*int)`      D. `(int *)`

## 共用体

22. 设有程序段

```
union data
{
 int i;
 char ch;
 double f;
}b;
```

设上面程序在 `VC++` 环境下运行, 则共用体变量 `b` 占用内存的字节数是 ( )。

- A. 1      B. 2      C. 8      D. 11

23. 设有程序段

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

```
typedef union
{
 long a[2];
 int b[4];
 char c[8];
}TY;
TY our;
int main()
{
 printf("%d\n", sizeof(our));
 return 0;
}
```

上面程序在 `VC++` 下运行, 输出的结果是 ( )。

- A. 32      B. 16      C. 8      D. 24

24. 当说明一个共用体变量时, 系统分配给他的内存是 ( )。

- A. 各成员所需内存量的总和  
B. 结构体中第一个成员所需的内存量  
C. 成员中占内存量最大者所需的容量  
D. 结构中最后一个成员所需内存量

25. C 语言共用体类型变量在程序运行期间 ( )。

- A. 所有成员一直驻留在内存中
- ☒ B. 只有一个成员驻留在内存中
- C. 部分成员驻留在内存中
- D. 没有成员驻留在内存中

26. 设有定义

```
union data
{
 int a;
 char b;
 double c;
}x, y;
```

若有以上程序片段，不正确的叙述是（ ）。

- A. 变量 x 所占内存的长度等于成员 c 的长度
- B. 变量 x 的地址和它的各成员地址都是相同的
- C. 可以在定义时对 x 初始化
- ☒ D. 以上说法都不对

27. 设有程序

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

```
union pw
{
 int i;
 char ch[2];
} a;
```

```
int main()
{
 a.ch[0] = 13;
 a.ch[1] = 0;
 printf("%d\n", a.i);
 return 0;
}
```

上面程序输出的结果是（ ）。

- ☒ A. 13
- B. 14
- C. 208
- D. 209

28. 设有定义

```
union data
{
 int i;
 char c;
 float f;
}a;
```

共用体变量不能整体操作

```
int n;
```

则以下语句正确的是（ ）。

- ☒ A. a.i = 5
- B. a = {2, 'a', 1.2};
- C. printf("%d\n", a);
- D. n = a;

## typedef 用法

29. 下面试图为 double 说明一个新类型名 real 的语句中，正确的是（ ）。
- A. typedef real double;
  - ☒ B. typedef double real;
  - C. typedef real = double ;
  - D. typedef double = real;

## 文件

30. C 语言可以处理的文件类型是（ ）。
- A. 文本文件和数据文件
  - B. 数据代码文件
  - ☒ C. 文本文件和二进制文件
  - D. 数据文件和文本文件
31. 下列对 C 语言的文件存取方式的叙述中，正确的是（ ）。
- A. 只能随机存取
  - B. 只能从文件的开头存取
  - C. 只能顺序存取
  - ☒ D. 可以顺序存取，也可以随机存取
32. 利用 fseek 函数可实现的操作（ ）。
- A. fseek (起始点, 位移量, 文件类型指针);
  - B. fseek (位移量, 起始点, fp);
  - C. fseek (文件类型指针, 起始点, 位移量);
  - ☒ D. fseek (fp, 位移量, 起始点);
33. 函数调用语句：fseek(fp, -20L, 2); 的含义是（ ）。
- A. 将文件位置指针移到离当前位置 20 个字节处
  - B. 将文件位置指针从当前位置向后移动 20 个字节
  - C. 将文件位置指针移到距离文件头 20 个字节处
  - ☒ D. 将文件位置指针从文件末尾处后退 20 个字节
34. 若执行 fopen 函数时发生错误，则函数的返回值是（ ）。
- A. 地址值
  - B. 1
  - C. EOF
  - ☒ D. 0
35. 设 fp 为指向某二进制文件的指针，且已读到此文件末尾，则函数 feof(fp) 的返回值为（ ）。
- A. EOF
  - ☒ B. 非 0 值
  - C. 0
  - D. NULL
36. fgetc 函数的作用是从指定文件读出一个字符，该文件的打开方式必须是（ ）。
- A. 只写
  - ☒ B. 读或读写
  - C. 追加
  - D. 读、读写或追加
37. 若要用 fopen 函数打开一个新的二进制文件，该文件要既能读也能写，则文件方式字符串应是（ ）。
- A. "rb+"
  - B. "ab+"
  - C. "ab"
  - ☒ D. "wb+"
38. 要打开一个已存在的非空文件 "file" 用于修改，选择正确的语句（ ）。
- A. fp = fopen("file", "r");
  - ☒ B. fp = fopen("file", "r+");
  - C. fp = fopen("file", "a+"); 错误

|            |
|------------|
| SEEK_SET 0 |
| SEEK_CUR 1 |
| SEEK_END 2 |



D. fp = fopen("file", "w");

## 二、程序题目

以下是函数这章练习题目

结构体、文件这两章一定要做实验手册的习题  
程序题有链表操作

课后作业 1

程序作业 第 18、19 次作业