

# 21 年 C 语言期末试题

严正声明: 本文内的代码如果引起不适, 与编写者没有半毛钱关系, 本人绝对不会像这里一样写代码.

一些实在看不过去的(比如忽略函数的返回类型与错误缩进), 在不影响题目考察内容的情况下已被修正, 不太过分的也已保留以便学弟学妹们感受风格.

请千万不要像这试卷一样写代码, 求求了.

## 一. 选择题

1

```
struct test {  
    int i;  
    char c;  
    float a;  
} test;
```

求 `sizeof(test)`.

- A. 6
- B. 7
- C. 9
- D. 与平台/编译器相关

2

在 C 语言里, 下列说法正确的是:

- A. 每一行只能由一条语句组成
- B. 函数的形参与实参共同使用同一个储存单元
- C. 整型与实型均能被准确表示
- D. `%` 仅用于整型求余

3

下列声明错误的是:

- A. `int a[] = {1, 2};`
- B. `char *a[3];`
- C. `char s[10] = "test";`
- D. `int n = 5, a[n];`

4

下列对外部二维数组初始化正确的是:

- A. `int a[2][] = {{1, 0, 1}, {5, 2, 3}};`
- B. `int a[][3] = {{1, 2, 1}, {5, 2, 3}};`
- C. `int a[2][4] = {{1, 2, 1}, {5, 2}, {6}};`
- D. `int a[][3] = {{1, 0, 2}, {}, {2, 3}};`

## 5

若有 `float a = 2.5`, 那么 `a * 3.14` 的类型是?

- A. `float`
- B. `double`
- C. `long double`
- D. `long int`

## 6

下列初始化错误的是:

- A. `char str[6] = {'a', 'h', 'a', 'h', 'a', '\0'};`
- B. `char *str = {'a', 'h', 'a', 'h', 'a'};`
- C. `char str[] = {"ahaha"};`
- D. `char str[] = "ahaha";`

## 7

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

下列说法错误的是:

- A. `struct` 是用于声明结构体的关键字
- B. `struct stu` 是一个结构体类型名
- C. `stutype` 是一个用户定义的结构体的类型
- D. `a` 和 `b` 都是结构体类型 `struct stu` 的成员

## 8

```
int x; float y;  
scanf("x=%d,y=%f", &x, &y);
```

若要使得 `x` 与 `y` 分别被赋值为 `10` 与 `66.6`, 应该输入什么?

- A. `x=10,y=66.6<回车>`
- B. `10,66.6<回车>`
- C. `10 66.6<回车>`
- D. `x=10 y=66.6<回车>`

## 9

```
#include <stdio.h>  
int main() {  
    int a = 1, b = 2, c = 3;  
    if (a > b)  
        if (a > c)  
            printf("%d", a);  
    else  
        printf("%d", b);  
    printf("%d", c);  
    return 0;  
}
```

上述程序的输出是?

- A. `1 2`

- B. 2 3
- C. 3
- D. 上面的都不对

## 10

已知数组声明 `int a[3][4];`, 下面哪个选项不能访问到数组的第 `i` 行第 `j` 列个元素?

- A. `a[i][j]`
- B. `*(*(a + i) + j)`
- C. `*(a[i] + j)`
- D. `*(a + i*4 + j)`

## 11

```
#include <stdio.h>
int main() {
    int a = 5, b = 4, x, y;
    x = 2 * a++;
    printf("a=%d, x=%d\n", a, x);
    y = --b * 2;
    printf("b=%d, y=%d\n", b, y);
    return 0;
}
```

求程序的输出.

- A. `a=6, x=12` 下一行 `b=3, y=6`
- B. `a=6, x=10` 下一行 `b=3, y=6`
- C. `a=6, x=10` 下一行 `b=3, y=8`
- D. 以上都不对

## 12

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

void amovep(int *p, int (*a)[3], int n) {
    int i, j;
    for (i = 0; i < 3; i++)
        for (j = 0; j < n; j++) {
            *p = a[i][j];
            p++;
        }
}

int main() {
    int a[3][3] = {{1, 3, 5}, {2, 4, 6}}, *p;
    p = (int*)malloc(100);
    amovep(p, a, 3);
    printf("%d, %d", p[2], p[5]);
    free(p);
    return 0;
}
```

求程序的输出.

选项忘记了

## 13

```
#include <stdio.h>

int f() {
    static int i = 0;
    int s = 1;

    s += i;
    i++;
    return s;
}

int main() {
    int i, a = 0;
    for (i = 0; i < 5; i++) a += f();
    printf("%d", a);
    return 0;
}
```

求程序的输出.

- A. 20
- B. 24
- C. 25
- D. 15

## 14

若要以可读写模式打开位于 A 盘 user 目录下的 abc.txt, 对应的 C 代码是:

选项顺序可能不正确

- A. `fopen("A:\user\abc.txt", "r");`
- B. `fopen("A:\\user\\abc.txt", "r+");`
- C. `fopen("A:\user\abc.txt", "rb");`
- D. `fopen("A:\\user\\abc.txt", "w");`

## 15

```
#include <stdio.h>

void ss(char *s, char t) {
    while (*s) {
        if (*s == t)
            *s = t - 'a' + 'A';
        s++;
    }
}

int main() {
    char s[100] = "abcddfefdbd", c = 'd';
    ss(s, c);
    printf("%s", s);
    return 0;
}
```

求程序的输出.

选项忘记了

## 16

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

struct NODE {
    int num;
    struct NODE *next;
};

int main() {
    struct NODE *p, *q, *r;
    p = (struct NODE *)malloc(sizeof(struct NODE));
    q = (struct NODE *)malloc(sizeof(struct NODE));
    r = (struct NODE *)malloc(sizeof(struct NODE));
    p->num = 10;
    q->num = 20;
    r->num = 30;
    p->next = q;
    q->next = r;
    printf("%d", p->num + q->next->num);
    return 0;
}
```

求程序的输出.

- A. 10
- B. 20
- C. 30
- D. 40

## 17

```
#include <stdio.h>

void fun(char *a, char *b) {
    a = b;
    (*a)++;
}

int main() {
    char c1 = 'A', c2 = 'a', *p1 = &c1, *p2 = &c2;
    fun(p1, p2);
    printf("%c%c", c1, c2);
    return 0;
}
```

求程序的输出.

考试时此题有两个相同选项, 已改正

- A. Ab
- B. Aa
- C. Bb
- D. 以上都不是

## 18

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

```

char *scmp(char *a, char *b) {
    if (strcmp(a, b) < 0)
        return a;
    return b;
}

int main() {
    char a[100], b[100], c[100], string[100];
    gets(a);
    gets(b);
    gets(c);
    strcpy(string, scmp(a, b));
    strcpy(string, scmp(string, c));
    printf("%s", string);
    return 0;
}

```

输入 `abcd` , `abba` , `abc` , 程序的输出是?

- A. `abcd`
- B. `abba`
- C. `abc`
- D. 以上都不对

## 19

```

struct Node {
    int data;
    struct Node *next;
};

```

已知有链表定义如上, `p` , `q` , `r` 分别指向该链表里连续的三个节点. 现想调换 `q` , `r` 所指向的节点的顺序, 同时保持链表连续, 则下列操作错误的是:

- A. `r->next = q; q->next = r->next; p->next = r;`
- B. `q->next = r->next; p->next = r; r->next = q;`
- C. `p->next = r; q->next = r->next; r->next = q;`
- D. `q->next = r->next; r->next = q; p->next = r;`

## 20

已知: `int a[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}`, `*p = &a[3]`; 求 `p[5]` 的值.

- A. 5
- B. 6
- C. 8
- D. 9

## 21

```

#include <stdio.h>

void f(int u, int v) {
    int w;
    w = u;
    u = v;
    v = w;
}

int main() {
    int x = 1, y = 3, z = 2;
}

```

```

    if (x > y) f(x, y);
    if (x > z) f(x, z);
    f(z, y);
    printf("x=%d, y=%d, z=%d", x, y, z);
    return 0;
}

```

求程序的输出.

选项忘记了

## 22

```

#include <stdio.h>

int a = 2;

int f(int *A) {
    return (*A)++;
}

int main() {
    int s = 0;
    { int a = 5;
      s += f(&a);
    }
    s += f(&a);
    printf("%d", s);
    return 0;
}

```

求程序的输出.

选项忘记了

## 23

```
void *func();
```

上述代码的意义是:

选项顺序不一定正确, 错误选项不一定完全与原卷一样

- A. `func` 没有返回值
- B. `func` 的返回值是无类型
- C. `func` 的返回值是一个无值型的指针
- D. `func` 是一个函数指针

## 24

```

struct student {
    int age;
    char num[8];
};

// 回忆者注: 此处原卷中 "19" 应是误作 10\9, 已修正
struct stu[3] = {{21, "200401"}, {20, "200402"}, {19, "200403"}}, *p = stu;

```

下列访问合法的是:

- A. `(p++)->num`
- B. `p->num`

- C. `(*p).num`
- D. `stu[3].age`

25

```
#include <stdio.h>

int main() {
    int y = 9;
    for (; y > 0; y--) {
        if (y % 3 == 0) {
            printf("%d", --y);
            continue;
        }
    }
    return 0;
}
```

程序的输出是?

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

## 二. 程序填空

主观题题面均不保证与试卷完全相同, 但本人将尽量保证其意思与试卷相同

主观题的代码可能跟试卷上的有稍微出入, 但本人可以保证没有考点/结构/主要(变量/函数)命名上的出入

1

下面的程序根据用户输入的字母建立反向链表, 借此做到逆转用户输入的目的. 例如, 用户输入 `abc`, 程序中建立起 `c -> b -> a` 的链表.

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

typedef struct {
    char data;
    struct Node *link;
} node;

node *head;

void ins(node /* 空 (1) */) {
    if (head == NULL) {
        head = q;
        q->link = NULL;
    } else {
        q->link = /* 空 (2) */;
        head = q;
    }
}

int main() {
    node *p;
    char ch;
```



```

while ((ch = getchar()) != EOF) {
    p = (node *)malloc(sizeof(node));
    p->data = ch;
    ins(/* 空 (3) */);
}

for (p = head; p; p = p->link)
    printf("%c", p->data);

return 0;
}

```

## 2

下面的程序输出个位, 十位, 百位都相同的三位数.

```

#include <stdio.h>
int main() {
    int g, s, b;
    for (g = 0; g < 10; g++) {
        for (s = 0; s < 10; s++) {
            for (b = 0; b < 10; b++) {
                if (/* 空 (1) */)
                    printf("%d", /* 空 (2) */);
            }
        }
    }
    return 0;
}

```

## 3

假如你一开始有 1020 个果子, 第一天卖掉一半多两个, 第二天卖掉剩下的一半多两个, 以此类推, 你要卖几天? 下面的程序解决了这个问题:

```

#include <stdio.h>
int main() {
    int x1 = 1020, x2, day = 0;
    while (/* 空 (1) */) {
        x2 = /* 空 (2) */;
        x1 = x2;
        /* 空 (3) */;
    }

    printf("%d", day);
    return 0;
}

```

## 4

下面的程序对用户输入的字符串里的字符 'a' 计数:

```

int main() {
    char s[100], *p;
    int count;

    scanf("%s", s);

    /* 空 (1) */;
    for (/* 空 (2) */; *p; p++) {

```

```

        if (*p != 'a')
            continue;
        /* 空 (3) */;
    }

    return 0;
}

```

## 5

下面的函数计算  $x$  与  $y$  的和, 并把结果通过指针形参 `sum` 传出:

```

void Add(int x, int y, /* 空 (1) */) {
    /* 空 (2) */ = x + y;
}

```

## 6

下面的程序求解不等式  $1^3 + 2^3 + \dots + m^3 < n$ , 其中  $^$  表示幂. 请注意 `n` 的类型为 `long`.

```

#include <stdio.h>
int main() {
    long n, m, i = 1, sum = 0;
    scanf(/* 空 (1) */);

    for (i=1; ;i++) {
        sum += /* 空 (2) */;
        if (/* 空 (3) */) break;
    }

    m = i - 1;
    printf("m ≤ %ld", m);
    return 0;
}

```

## 7

下面的程序让用户输入一个字符串, 随后在字符串里的每一个字符之间都插入一个空格并输出:

```

#include <stdio.h>

#define MAX 100
/* 空 (1) */

void Insert(char strSrc[]);

int main() {
    char string[MAX];
    scanf("%s", string);
    Insert(string);
    printf("%s", string);
    return 0;
}

void CopyString(char c[], char s[]) {
    int i;
    for (i = 0; s[i]; i++)
        c[i] = s[i];
    /* 空 (2) */;
}

```

```

}

void Insert(char strSrc[]) {
    int i = 0, j = 0;
    char strTemp[MAX];
    CopyString(strTemp, strSrc);

    while (/* 空 (3) */) {
        strSrc[i] = strTemp[j];
        i++;
        /* 空 (4) */;
        strSrc[i] = ' ';
        i++;
    }
}

```

## 8

下面的程序让用户输入矩阵, 并且计算矩阵里每一列的最小值之和:

```

#include <stdio.h>
#define M 4
#define N 2

int SumColumnMin(int a[M][N], int *sum) {
    int i, j, k, s = 0;
    for (i = 0; i < M; i++) {
        k = 0;
        for (j = 0; j < N; j++) {
            if (a[j][i] < a[k][i])
                k = j;
        }
        /* 空 (1) */;
    }
    /* 空 (2) */;
}

int main() {
    int x[M][N], i, j, s;

    for (i = 0; i < M; i++) {
        for (j = 0; j < N; j++)
            scanf("%d", &x[i][j]);
    }

    SumColumnMin(/* 空 (3) */);
    printf("%d", s);
    return 0;
}

```

## 9

下面的程序让用户输入字符, 并且判断该字符是否为大小写字母, 数字, 空格或其他:

```

#include <stdio.h>

int main() {
    char ch;
    ch = getchar();
}

```

```
if (/* 空 (1) */) {
    puts("This is an English character.");
} else if (/* 空 (2) */) {
    puts("This is a digit.");
} else if (/* 空 (3) */) {
    puts("This is a space.");
} else {
    puts("This is other character.");
}

return 0;
}
```

### 三. 手写程序题

你将要实现一个程序, 其接收用户输入的字符串, 统计字符串内数字字符的总数量, 并分别计算各个数字字符的数量, 然后将二者输出.

你的程序里应该按下面原型与描述实现两个函数:

```
// 统计字符串内数字字符的数量, 并返回它
int DigitBit(char str[]);

// 统计并输出字符串内各个数字字符的数量
void DigitCount(char a[]);
```

输入提示符: "Input a number:"

输入格式: "%s"

总数字字符数输出格式: "%d bits\n"

数字字符分别数量输出格式: "%d: %d\n"

样例输入输出如下:

```
Input a number: 12223
5 bits
1: 1
2: 3
3: 1
```

```
Input a number: -12asdfn666
5 bits
1: 1
2: 1
6: 3
```