

一、看代码写结果

要做对这些题，不仅要掌握题目考察的语法知识，还要细心模拟。

如果脑内算力不足，最好打草稿，每次变量变化了记一下，每个语句的范围圈一下。

1

逗号表达式的结果

逻辑表达式的结果要么是0，要么是1

`||` 是短路运算符，并且两个都是0才为0

`+=`、`/=` 是右结合、`+` 是左结合，[参考文章](#)

2

`%lf` 默认输出小数点后六位

用 `int` 给浮点数做强制类型转化的方式是截断，也就是直接去掉小数部分

`%x` 是以十六进制形式输出无符号整数(不输出前缀0x)，并且十六进制的A到F是大写，[更多printf的知识](#)

C语言，整数除法的结果也是对应的浮点数结果去掉小数部分，比如 $33/4=8.25 \Rightarrow 8$ ， $-33/4=-8.25 \Rightarrow -8$

`'a'+4`用`printf %c`输出是字符e

3

x是十六进制，每位十六进制对应四位二进制，这里就是001000000010001100010001

用char对x做强制类型转化就是截断，char是1个字节即8位，`(char)x`就是取x的低8位，也就是二进制00010001，用`%d`输出转为十进制就是17

`char a = -128`，二进制为8位的10000000，如果看成32位的int，就是11111111 11111111 11111111 10000000，是带符号扩展。

那`a--`就是`a-1`，也就是加上-1的补码11111111，相加的结果就是00000000 00000000 00000000 01111111，用`%d`输出就是127

`'x'-'Z'+127`的计算方法：首先要知道`'a'-'A'=32`，那么`'x'-'Z'+127='a'+23-('A'+25)+127=157`

4

要知道if语句后面接一个语句块，一对if-else算一个语句块，一个大括号里的所有语句算一个语句块。

`&&` 是短路运算符，前一个结果为假，后面的不会执行

5

`++i` 是先加再用

注意for循环每次i+=2

`if(!(++i%2))` 等价于 `if ((++i) % 2 == 0)`

6

`p[i]++` 是先用再加

要注意switch-case语句的执行顺序: [参考文章](#)

```
1 switch(表达式){
2     case 整型数值1: 语句 1;
3     case 整型数值2: 语句 2;
4     .....
5     case 整型数值n: 语句 n;
6     default: 语句 n+1;
7 }
```

它的执行过程是:

- 1) 首先计算“表达式”的值, 假设为 m 。
- 2) 从第一个 case 开始, 比较“整型数值1”和 m , 如果它们相等, 就执行冒号后面的所有语句, 也就是从“语句1”一直执行到“语句n+1”, 而不管后面的 case 是否匹配成功。
- 3) 如果“整型数值1”和 m 不相等, 就跳过冒号后面的“语句1”, 继续比较第二个 case、第三个 case.....一旦发现和某个整型数值相等了, 就会执行后面所有的语句。假设 m 和“整型数值5”相等, 那么就会从“语句5”一直执行到“语句n+1”。
- 4) 如果直到最后一个“整型数值n”都没有找到相等的值, 那么就执行 default 后的“语句 n+1”。

也就是说, 不是default写在case前面就会先执行default, 应该是先执行相匹配的case, 如果没有匹配的才执行default, 或者default写在匹配的条件后面被顺序执行了

```
1 // 比如这里, case 1进去执行i++, 顺序执行default的i++和case2的i++
2 int i = 1;
3 switch (i)
4 {
5     case 1:
6         i++;
7
8     default:
9         i ++ ;
10    case 2:
11        i ++;
12    }
13    printf("%d", i);
```

还要注意的, switch里的break的效果是跳出switch, 不是跳出while循环。switch语句的语法里有break, 没有continue, 所以continue就是为循环控制服务的

7

`-.1e-2`就是`-0.1e-2 == -0.001`

三目运算符 `a?b:c`, 如果a为真, 执行b, 否则执行c

8

注意每次j的值不要算错了

$N = 7, N/2 = 3$

$i = 0, j = 6$

$i = 1, j = 5$

$i = 2, j = 4$

9

这里的知识点是转义字符 `\x` 的规则，[参考文章链接](#)

具体到本题，`\x` 后面最多跟2个数字，那这里就是 `\x86`，它们在一起相当于一个字符。所以字符串里有 `w`、`i`、`\n`、`\x86` 四个字符。

经过我上机实践，发现 `\x` 会把它后面连续的数字都做转义，并且只取最后两位数，其他的不起作用

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main()
5 {
6     char s[] = "win\x86841";
7     printf("%d\n%s", strlen(s), s);
8     // 输出winA
9     // 即\x86841相当于A
10    // 'A'的ascii码是65，转为十六进制为41，所以'A'的十六进制表示就是'\x41'，可见中间
    // 的868被舍弃掉了
11    return 0;
12 }
```

10

要注意二维数组的存储方式和初始化

```
1 char s[][10] = {"1234", "56789"};
2 // 定义的数组是
3 s[0]: 1 2 3 4 0 0 0 0 0 0
4 s[1]: 5 6 7 8 9 0 0 0 0 0
```

二、改错题

错误包含两种：

1. 语法错误
2. 逻辑错误

不能只找语法错误，而不去理解代码是怎么解决题干中的问题的。

语法错误：

1. `#define`和`#include`后面不能有 `;` 号

2. `scanf %d、%c、%lf`要记得取地址符号 `&`
3. `&&` 是逻辑运算符, `&` 是位运算中的按位与, `||` 和 `|` 同理
4. `for`循环和`while` (不是`do-while`)后面一般不会直接加`;`, 除非是故意这样写的。
5. `=` 和 `==` 不要搞混了

逻辑错误:

1. 有的时候数组`#define`的数组长度可能不够长, 不符合题目要求
2. 注意 `>`、`>=`、`<`、`<=` 有没有写对
3. 循环里什么时候应该写`continue`, 什么时候应该写`break`
4. `switch`语句里是不是漏写了`break`
5. `for`循环和`while`循环的起始和终止有没有写对, 注意`i < n`说明`i`最大到`n-1`
6. 数组下标不要越界

三、编程题

个人解答, 最好看答案吧

第一题

```
1  #include <stdio.h>
2
3  int check(int x)
4  {
5      int sum = 0, t;
6      while (x)
7      {
8          t = x % 10;
9          x /= 10;
10         if (t == 8) return 1;
11         else sum += t;
12     }
13     return sum == 8;
14 }
15
16 int main()
17 {
18     int M, N;
19     scanf("%d%d", &M, &N);
20     for (int i = M; i <= N; i ++ )
21     {
22         if (check(i))
23             printf("%d ", i);
24     }
25     return 0;
26 }
```

第二题

```
1  #include <stdio.h>
2  #include <ctype.h>
3
4  #define N 110
5
```

```

6  int main()
7  {
8      char a[N], b[N];
9      scanf("%s%s", a, b);
10     for (int i = 0; a[i]; i ++ )
11     {
12         if (isupper(a[i]))
13             a[i] += 'a' - 'A';
14     }
15     for (int i = 0; b[i]; i ++ )
16     {
17         if (isupper(b[i]))
18             b[i] += 'a' - 'A';
19     }
20     int res = 0;
21     for (int i = 0; a[i]; i ++ )
22     {
23         if (a[i] == b[0])
24         {
25             int k = 0;
26             for (int j = i; a[j] && b[k]; j ++ , k ++ )
27             {
28                 if (a[j] != b[k])
29                     break;
30             }
31             if (!b[k]) res ++ ;
32         }
33     }
34     printf("%d", res);
35     return 0;
36 }

```

第三题

```

1  #include <stdio.h>
2  #include <math.h>
3
4  #define N 2010
5
6  int main()
7  {
8      int st[N] = {0};
9      int n, m;
10     scanf("%d%d", &n, &m);
11     double a;
12     int t, ta;
13     for (int i = 0; i < m; i ++ )
14     {
15         scanf("%lf%d", &a, &t);
16         for (int j = 1; j <= t && floor(j * a) <= n; j ++ )
17         {
18             int ind = floor(j * a);
19             st[ind] = !st[ind];
20         }
21     }
22     for (int i = 1; i <= n; i ++ )

```

```

23     {
24         if (st[i])
25             printf("%d ", i);
26     }
27     return 0;
28 }

```

答案

参考答案

一、(40 分) 评分标准：每小题 4 分

题号	答案
1	32, 0, 1, 0
2	d=19.500000, i=5 j=21, e
3	17 127, 157
4	2, 1, 3, 6
5	2, 6
6	6, 10, 2, 12
7	10.00 8.63
8	7 6 5 4 3 2 1
9	4 0 Radeon i Radeon R7
10	23 4 679 8

二、改错题 (20 分)

评分标准：共 11 个错，改对 1 个加 2 分（即允许有一个错误没发现，降低难度，满分 20）

1. 第 2 行：max 后多一个分号
2. 第 5 行：cnt 赋初值为 0
3. 第 8 行：a[i] 前面加上 &
4. 第 9 行：i-4 改为 i-5
5. 第 10 行：& 改为 &&
6. 第 14 行：continue 改为 break
7. 第 16 行：= 改为 ==
8. 第 19 行：j<cnt-1 改为 j<cnt 或 j<=cnt-1
9. 第 20 行：大于号改为小于号
10. 第 22 行与 23 行对调
11. 第 27 行：去掉 for 括号后的;

三、编程题（40 分）

1.

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

```
int main() {
    int start, end;
    scanf("%d %d", &start, &end); // 输入
    for (int i = start; i <= end; i++) { // 遍历范围
        int sum = 0, n = i, digit = 0; // n=i 避免运算后输出时 i 丢失
        while (n) { // 提取每个位上的数字
            digit = n % 10; // 提取单个位上的数字
            if (digit == 8) // 判断数字 8
                break;
            sum += digit;
            n /= 10; // 提取单个位上的数字
        }
        if (digit == 8 || sum == 8) // 判断数字 8 或者各位数和为 8
            printf("%d ", i);
    }
    printf("\n");
    return 0;
}
```

注意事项:

- (1) 未正确包含头文件，如使用数学函数必须包含 `math.h`;
- (2) 未正确处理输入输出，如 `scanf` 缺少 `&` 符号，格式说明符不匹配数据类型;
- (3) 未正确书写 `for` 循环，如 `<=end` 写为 `<end`;
- (4) 未正确提取数字，未提取单个位的数字;
- (5) 未正确循环提取每个位的数字（初始状态、循环条件、状态改变错）;
- (6) 直接在 `i` 上修改导致运算后 `i` 丢失的;
- (7) 未正确判断数字 8，含判断正确未跳出循环;
- (8) 未正确判断各位数字和为 8;
- (9) 未正确判断输出条件，如 `||` 写成 `&&` ;
- (10) 未正确书写 `main` 函数，或写错返回语句;
- (11) 声明语句错，如 `start,end,i` 等精度不小于 `int`（可 `long long, long double`）。

2.

```
#include <stdio.h>
int main() {
    char a[101], b[101]; // 字符串有 100 个字符且串尾为 0, 长至少 101
    scanf("%s%s", a, b); // 输入 AB, 不取地址
    int i, j, k, occurrence = 0; // 记录出现次数应初始化为 0
    for (i = 0; a[i] != '\0'; i++) { // 将 AB 的同置为大写
        if (a[i] >= 'a')
            a[i] -= 'a' - 'A'; // 先求差, 再转换, 避免溢出
    }
    for (i = 0; b[i] != '\0'; i++) {
        if (b[i] >= 'a')
            b[i] -= 'a' - 'A';
    }
    for (i = 0; a[i] != '\0'; i++) {
        int isfound = 1; // 假设已找到, 发现不匹配即置为未找到
        // 在 A 当中查找 B, 在 A 或 B 之一达到串尾即退出循环
        for (j = i, k = 0; a[j] != '\0' && b[k] != '\0'; j++, k++) {
            if (a[j] != b[k]) { // 一旦出现字母不匹配, 判定失败
                isfound = 0;
                break; // 判断失败退出循环, 未退出也可以
            }
        }
        // 退出是因为找到差异后 break, 还是没找完字符串 A 已达串尾
        if (isfound == 1 && b[k] == '\0') { // b[k] == '\0' 确保字符串 B 完整遍历
            occurrence++;
        }
    }
    printf("%d\n", occurrence);
    return 0;
}
```

注意事项:

- (1) 整体上, 输入, 大小写无关, 判断 B 在 A 中的功能, 输出;
- (2) 未字符串 A,B 进行大小写无关的比较; 同转为大写或者同转为小写, 都正确;
- (3) 使用 string.h, 或者 strlen, strcmp, strstr, substr, strcat 等函数的;
- (4) 无法判断遍历时 A 已结束但 B 未结束的情况, 即 aais 中找 saa 的个数结果为 1 (即只判断了字母 s 就判定成功) 的;
- (5) 循环条件 a[j] != '\0' && b[k] != '\0', && 写为 || 或遗漏条件的; 其中, '\0' 与不带单引号的 0 是等价写法。
- (6) 初始化错误, 如: occurrence 未初始化为 0, isfound 未在新一轮循环初始化为 1;
- (7) 开辟 A 和 B 空间少于 101 个元素的, scanf 或 gets 处误对字符串用取地址符号的 (如 a 写为 &a)。
- (8) 未能正确使用 #include 指令, main 函数等, 或其它小的编译错误。

3.

```
#include <stdio.h>
int main() {
    int s[2001] = { 0 }, m, n; // 本例使用 1..2000 故应为 2001
    scanf("%d %d", &m, &n); // 接受输入灯数和以下行数
    for (int i = 1; i <= n; i++) { // 遍历接受每一行输入
        float a;
        int t;
        scanf("%f %d", &a, &t); // 接受输入 a 和 t
    }
}
```



```

        for (int j = 1; j <= t; ++j) {    // 遍历 1 到 t 用于计算灯号
            int pos = (int)(j * a);        // 向下取整得到灯号
            if (s[pos] == 0)                // 切换亮灯状态，可以有其它等价写法
                s[pos] = 1;                // 写为 s[pos]=(s[pos]==0)
            else                            // 或者 s[pos]=1-s[pos] 也可以
                s[pos] = 0;
        }
    }
    for (int i = 1; i <= m; i++) {        // 依次遍历灯状态，输出亮灯号
        if (s[i] == 1) {                  // 判断灯亮
            printf("%d ", i);
        }
    }
    printf("\n");
    return 0;
}

```

注意事项:

- (1) 整体上，输入，开关灯功能，输出；
- (2) 输入语句，未使用取地址符号&的，格式说明符和声明的数据类型不符的，每处错误；
- (3) 输出语句，判断灯亮错误的，或循环的条件错误的（如<=m 写为<m 的），格式说明符错的。
- (4) 开关灯功能循环的条件错误的（如<=t 写为<t 的）。
- (5) 未正确切换亮灯状态的，；注意不必限制 1 表示亮 0 表示暗，其它方案只要前后一致且不溢出即可；
- (6) 计算亮灯号错误的，如未下取整的，扣 2 分，但在该例中如果 pos 为整型，强制类型转换不是必须的。
- (7) 数据类型错误的（选用类型比题中范围更大，不记为错误），声明数组未初始化的，或声明数组长度小于 2000（如下标 0 不使用则应为 2001，具体以 N=2000 时是否越界为准）；
- (8) 未能正确使用#include 指令（如使用 floor()函数必须包含 math.h），main 函数等，或其它小的编译错误，每处错误。