电子系程设期末笔试题答案解析

2020年1月5日

1.20 29

首先不妨将数组写成如下等价形式：

```
int a[3][3] = {
    {1,  3,  5 },
    {7,  9,  11},
    {13, 15, 19}
};
```

对于sum1，由于`i==j`约束，因此sum1= a[0][0] + a[1][1] + a[2][2] = 1 + 9 + 19 = 29;

对于sum2，由于`i+j==2`约束，因此sum2 = a[0][0] + a[1][1] + a[2][2] = 5 + 9 + 13 = 27。

注意最后15后是19，毕竟本质上算的是两条对角线分别求和，如果真的放等差数列的话两个求和就会相同了。

2.10,20,x 50,60,c

函数`f1(struct p b)`的传参方式是复制了结构体进行传递，此时内部的修改仅仅修改了一个临时变量b自己的内容，对外界没有影响，所以执行完`f1(a)`后结构体a没有改变。

函数`f2(int x[2])`本质上传递了一个`int *`，因此执行`f2(a.x)`时，函数通过该指针存储的地址访问了结构体a的内存，实际地改变了a的内容。

3.HOW how do you do DO YOU DO

最初三个指针p1, p2, p3分别指向了三个字符串str1, str2, str3。

`scanf("%s", p2)`使得字符串str2读取一串内容，输入字符串时遇到空格断开，实际读入了`HOW`。

`gets(p3)`执行时从之前`HOW`后面的空格开始，一直读到了回车，因此str3实际读到了`DO YOU DO\n`。

因此写答案时`do`与`DO`之间有两个空格。

4.13

虽然题目写得比较曲折，但是稍作观察不难发现，其实是在用递归法求斐波那契数列，fib(7)=13。

5.18

最初指针p指向数组a，k指向p，此时*p==a[0]==3，因此z第一次赋值后为3。

之后p=p+1，因此**k == *(p + 1) == p[1] == 15，故z = 3 + 15 = 18。

6.24

此处需要注意`f((x++, y--, x + y), z--)`这个表达式，首先其中的`(x++, y--, x + y)`是用了逗号表达式，真实的值是`x + y`，此前`x++`和`y--`已经执行，此处可能涉及到知识盲区的是`x + y`的x和y究竟是自加前的还是自加后的？所幸出题人并不打算考察这一点，无论先后最终加加减减抵消，只需要知道逗号表达式特性即可。

实际上自加和自减会先执行，传参是f((6+9), 9)，所以返回值是6+9+9=24。

7.1 2 3 4 5

注意k++是在该行之后加以及switch不break的后果即可，照着执行自然能得到答案。

8.5

细心地照着执行即可。

i	n[0]	n[1]	n[2]
初始化前	0	0	0
0	0+1=1	1+1=2	2+1=3
1	2+1=3	2+1=3	3+1=4
2	4+1=5	4+1=5	4+1=5

9.11 13 15

主函数种的static对于做题没有影响，但是`f(int a)`种的static就要注意了。实际运行是第一次执行f(a++)时返回的是2+6+3=11，考虑到之后每次a与c均会+1，b不变，所以直接写出后面两项答案是11+2=13，13+2=15。

10.23

注意B(a+b)展开后是`4*a+b/2`，故c = 4 * 5 + 6 / 2 = 23。

11.43

14年原题，实际上定义了两个链表结点，并且相互用指针p指向了对方。

最终h[0].p->x == h[1].x == 4, h[1].p->y == h[0].y == 3。

12.0 6

*p1==8, m == 6, 由于8!=6，故a == 0。

`(-*p1)/(*p2)`即-8 / 6 = -1，故b = -1 + 7 = 6。

13.5,6.000000

14年原题，实际上对数组a的偶数进行计数与求和，最后再用ave记录均值，实际上一共有5个偶数，均值是6，刚好除得尽故不涉及整数计算的舍入，输出时需要注意%f默认会带6位小数。

（这道题真的出得很*，我不知道为什么会再次考出来）

14.3 4 5 6

经过前面一轮赋值后此时的p完全可以当a用(尽管它们有本质差异)，另一方面`*(*(p + i) + j)`也即`p[i][j]`，不要疏忽了i只取了1和2，因此输出的是`a[1][0] a[1][1] a[2][0] a[2][1]`。

15.1 10 0 0 20 30

首先创建了文件"a.dat"并且写入了四个int`{1,10,20,30}`，之后用rewind切回文件开头，先读入两个int到b，这使得b[0]=1, b[1] = 10。再读入两个int到&b[4]，这使得b[4] = 20, b[5] = 30。

16.7 10

不少同学在考场上注意到了这是一道运行时数组越界的题……

由于传入了a+1，因此实际上函数内的a是`{{2, 4, 6}, {7, 8, 9}}`，p所指的内存变成了`{3, 5, 7, 8, 9, 10}`。

17.5,21,hahaa

`sizeof(st)`毫无悬念是21，因为`char st[21]`已经决定了。然后需要注意字符串内部有一个`'\0'`，故此时st只看得见前五个字符组成的字符串。

思考：如果把st的定义改成`char st[]="hahaa\0";`，答案会是什么？

18.7

我在讲课时曾经说过对于`a[2][3]={1,2,3,4,5,6}`，可以通过`a[0][i]`(i=0,1,2,3,4,5)直接访问整个数组，这里就属于这种访问法。

注意到`*(*p+i)`其实就是`p[0][i]`，因此p最初指向数组a，然后将12个数字遍历了一遍，找出了最大值7。

19.

DFKP

实际上输出了字符`a[0][0], a[1][1], a[2][2], a[3][3]`，得到答案。

注：

1. 据说根据考试要求，换行不用体现，空格有所体现即可，因此末尾空格不用担心，第3题中的连续两个空格我也不确定是否需要体现。
2. 刚考完的同学如果发现答案和考场写的答案不一致，不必担心，因为可能题目复现有误。
3. 第13题涉及到浮点数ave的输出，我不知道C++版怎么写的，如果是`cout << ave;`，那么答案就是6。