

# 第 13 讲

授课教师：郑炜

厦门大学计算机系



# 如何正确看待期中考试的结果？

发信人: PzkpfwV (再重申一遍我不是重型坦克~), 信区: CProgramming  
标题: Re: 我终于要下决心开除那个不称职的属下了  
发信站: 水木社区 (Sat Sep 4 20:09:42 2010), 站内

作为70后，我来给您说个故事吧~

以前大学的时候，C语言课，班里考试第一名，就是这样的神人  
一个女生，把C语言当成一门真语言来学——额，我的意思是，  
你想象一下你怎么学英语的吧……

就见那女娃的课本上勾满重点，复习的时候如果在教室，就掏出草稿纸来默写语句和函数。当然，有时候她也不在教室的。不要以为她去机房实践了，她是去校园里的小树林，手持课本&课堂笔记大声背诵。除了在机房的课时，她从不自己去学校公共机房。宿舍当然也没有电脑，就这样，期末考了96分，绝杀……

后记：

此女差点儿拿到保研资格。毕设时原形毕露，在N个哥们帮助下才完成，她付出了请客20多次的代价（此处省略若干字。。。），毕业后去了山东一所二本学校教计算机。



# 看代码，写结果 (1)

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int x,y,z,w;
6      x=(y=8,z=16,w=32);
7      y=!z || w>z>y;
8      z=x==w;
9      w+=w/=w+1;
10     printf("%d,%d,%d,%d", x, y, z, w);
11
12     return 0;
13 }
```

g++ 10 ☒ Compiler options...

Program returned: 0

Program stdout

32,0,1,0



# 看代码，写结果（1）解析

```
① int x,y,z,w;  
② x=(y=8,z=16,w=32);  
③ y=!z || w>z>y;  
④ z=x==w;  
⑤ w+=w/=w+1;  
⑥ printf("%d,%d,%d,%d",  
    x, y, z, w);
```

第二行	赋值运算符右侧为逗号表达式，其值是最右的赋值语句的结果即32，同时使得y赋值为8，z赋值为16，w赋值为32，因此x赋值为32
第三行	赋值运算符右侧是或运算表达式，第一个操作数!z的结果为0，未发生短路运算，w>z>y按左结合处理，等价于1>y，不成立因此结果为0，因此y=0  0结果还是0
第四行	关系运算符的优先级高于赋值运算符，因此先有表达式x==w成立，结果为1，因此z赋值为1
第五行	复合赋值运算按右结合处理，先有w/=w+1，即w=32/33，因w是整型的，因此w=0，w+=0还是等于0



## 看代码，写结果 (2)

```
1  #include <stdio.h>
2
3  int main()
4  {
5      double d=19.5;
6      int i=(int)d%7,j=33;
7      printf("d=%lf,i=%i\n",d,i);
8      printf("j=%X,%c\n",j,'a'+j/8);
9
10     return 0;
11 }
```

g++ 10



Compiler options...

Program returned: 0

Program stdout

d=19.500000,i=5

j=21,e



# 看代码，写结果（2）解析

- ① **double d=19.5;**
- ② **int i=(int)d%7,j=33;**
- ③ **printf("d=%lf,i=%i\n",d,i);**
- ④ **printf("j=%X,%c\n",j,'a'+j/8);**

初始	变量d是double类型，值为19.5
第二行	整型变量i等于d强制转换为整型后对7求余数，结果为5，j的值为33
第三行	%lf输出浮点数默认保留小数点后面6位，%i与%d是一样的，因此输出 <b>d=19.500000,i=5&lt;换行&gt;</b>
第四行	%X表示输出整数的十六进制，%c将整数值视为字符型输出，j=33十六进制为21，j/8结果为4，‘a’+4结果是‘e’，因此输出 <b>j=21,e</b>



# 看代码，写结果 (3)

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int x=0x202311;
6      printf("%d\n", (char) x);
7      char a=-128, b='x' - 'Z'; a--;
8      printf("%d,%d", a, b+127);
9
10     return 0;
11 }
```

g++ 10



Compiler options...

Program returned: 0

Program stdout

17

127,157



# 看代码，写结果（3）解析

- ① `int x=0x202311;`
- ② `printf("%d\n",(char) x);`
- ③ `char a=-128,b='x'-'Z'; a--;`
- ④ `printf("%d,%d",a,b+127);`

初始	变量x是int类型，值为16进制的202311，每两位16进制数对应1个字节
第二行	将整型变量x强制转换为字符型，发生截断，保留最低位的一个字节，即16进制的11，用%d十进制输出得 17<换行>
第三行	变量a是有符号的字符型，最小负值-128，再减1发生溢出，得到最大正值即127，'x'-'Z'='x'-'X'-2=32-2=30
第四行	b+127=157，因此输出 127,157





# 看代码，写结果 (4)

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int a=1, b=3, c=5, d=4, x;
6      if (a<b)
7          if (c<d&&++a) x=1;
8      else
9          if (a<c||b--)
10             if (b<d&&c++) x=2;
11      else x=3;
12          else x=6;
13              else x=7;
14      printf("%d,%d,%d,%d", x, a, b, c);
15
16      return 0;
17 }
```

g++ 10



Compiler options...

Program returned: 0

Program stdout

2,1,3,6



# 看代码，写结果（4）解析

```
① int a=1, b=3, c=5, d=4, x;  
② if (a<b)  
③     if (c<d&&++a) x=1;  
④ else  
⑤     if (a<c || b--)  
⑥         if (b<d&&c++) x=2;  
⑦ else x=3;  
⑧     else x=6;  
⑨     else x=7;  
⑩ printf("%d,%d,%d,%d", x, a, b, c);
```

先整理代码如右:

```
① int a=1, b=3, c=5, d=4, x;  
② if (a<b){  
③     if (c<d&&++a) {  
④         x=1;  
⑤     }  
⑥     else if (a<c || b--){  
⑦         if (b<d&&c++)  
⑧             x=2;  
⑨         else  
⑩             x=3;  
⑪     }  
⑫     else {  
⑬         x=6;  
⑭     }  
⑮ }  
⑯ else {  
⑰     x=7;  
⑱ }  
⑲ printf("%d,%d,%d,%d", x, a, b, c);
```

初始	a=1,b=3,c=5,d=4
第二行	a<b即1<3成立，转第3行
第三行	与运算表达式的左半部分c<d不成立，发生短路运算，右半部分++a不处理，a仍为1，if判断不成立，转第6行
第六行	或运算表达式的左半部分a<c成立，发生短路运算，右半部分b--不处理，b仍为3，if判断成立，转第7行
第七行	与运算表达式的左半部分b<d成立，继续处理c++，c的值更新为6，与运算结果为1，if判断成立，转第8行
第八行	x赋值为2后结束if嵌套，转第19行
第十九行	输出 2,1,3,6



# 看代码，写结果 (5)

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int i;
6      for (i=1; ;i+=2)
7      {
8          if (!(++i%2))
9          {
10             printf("%d,",i);
11             continue;
12          }
13          if (++i%3==0)
14             break;
15      }
16      printf("%d",i);
17
18      return 0;
19 }
```

g++ 10



Compiler options...

Program returned: 0

Program stdout

2,6



# 看代码，写结果（5）解析

```
① int i;  
② for (i=1; ;i+=2)  
③ {  
④   if (!(++i%2))  
⑤   {  
⑥     printf("%d,",i);  
⑦     continue;  
⑧   }  
⑨   if (++i%3==0)  
⑩     break;  
⑪ }  
⑫ printf("%d",i);
```

初始	i=1
第一次循环	第4行，先执行i=i+1=2，然后计算!(2%2)结果为1，if判断成立，转第6行，输出 <b>2</b> ，再执行continue，跳过第9和第10行，i=i+2=4，开始下一次for循环
第二次循环	第4行，先执行i=i+1=5，计算!(5%2)结果为0，if判断不成立，转第9行，先计算i=i+1=6，6%3结果为0，if判断成立，执行break跳出for循环，转第12行，此时i的值为6
第十二行	输出 <b>6</b>



# 看代码，写结果 (6)

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int p[7]= {5,9,12,23,30}, i=0, k=0;
6      while (p[i]++%2)
7      {
8          printf("%d,",p[i]);
9          switch(p[i+3]%5)
10         {
11             default: i++; continue;
12             case 3:  k+=3;
13             case 1:  k+=p[++i]; break;
14         }
15     }
16     printf("%d,%d",i,k);
17
18     return 0;
19 }
```

g++ 10



Compiler options...

Program returned: 0

Program stdout

6,10,2,12



# 看代码，写结果（6）解析

```
① int p[7]= {5,9,12,23,30}, i=0, k=0;
② while (p[i]++%2)
③ {
④   printf("%d,",p[i]);
⑤   switch(p[i+3]%5)
⑥   {
⑦     default: i++; continue;
⑧     case 3: k+=3;
⑨     case 1: k+=p[++i]; break;
⑩   }
⑪ }
⑫ printf("%d,%d",i,k);
```

初始	p[0]=5,p[1]=9,p[2]=12,p[3]=23,p[4]=30,p[5]=0,p[6]=0,i=0,k=0
第一次循环	i=0，先计算p[0]%2=5%2=1，循环条件成立，进入while循环，并且p[0]=p[0]+1=6，因此第四行输出 6, switch判断p[0+3]%5=23%5=3，转第8行，k=0+3=3，因没有break，继续执行第9行，先处理i=i+1=1, k=k+p[1]=12.
第二次循环	i=1，先计算p[1]%2=9%2=1，循环条件成立，进入while循环并计算p[1]=p[1]+1=9+1=10。因此第四行输出 10, switch判断p[1+3]%5=30%5=0，转第7行，执行i=i+1=2，然后执行continue进行下一次while循环
第三次循环	i=2，计算p[2]%2=12%2=0，循环条件不成立，循环结束，转第12行
第12行	按前述分析，i的值为2，k的值为12，因此输出 2,12



# 看代码，写结果 (7)

```
1  #include <stdio.h>
2  #include <math.h>
3  int main()
4  {
5      double t=0.5;
6      do
7      {
8          printf("%.21f\n",t>0?10:8.63);
9          t*=-.1e-2;
10     }
11     while (fabs(t)>1e-6);
12
13
14     return 0;
15 }
```

g++ 10



Compiler options...

Program returned: 0

Program stdout

10.00

8.63



# 看代码，写结果（7）解析

```
① double t=0.5;  
② do  
③ {  
④ printf("%.2lf\n",t>0?10:8.63);  
⑤ t*=-.1e-2;  
⑥ }  
⑦ while (fabs(t)>1e-6);
```

初始	t=0.5
第一次循环	因为t>0成立，条件表达式返回结果10，%.2lf表示输出浮点数保留小数点后面两位数字，因此输出 10.00<换行>; t = t*-.1e-2 = 0.5*(-0.001) = -0.0005，fabs(t)=0.0005 > 1e-6 = 0.000001，因此do-while循环继续
第二次循环	因为t=-0.0005>0不成立，条件表达式返回结果8.63，因此输出 8.63<换行>; t=t*-.1e-2=-0.0005*(-0.001)=0.0000005，fabs(t)>1e-6不成立，因此do-while循环结束





# 看代码，写结果 (8)

```
1  #include <stdio.h>
2
3  int main()
4  {
5      #define N 7
6      int i,j,temp,a[N];
7      for (i=0;i<N;i++)
8          scanf("%d",&a[i]);
9      for (i=0;i<N/2;i++)
10     {
11         j=N-1-i;
12         temp=a[i];
13         a[i]=a[j];
14         a[j]=temp;
15     }
16     for(i=0;i<N;i++)
17         printf("%d",a[i]);
18
19     return 0;
20 }
```

g++ 10



Compiler options...

1 2 3 4 5 6 7

Program returned: 0

Program stdout

7654321



```
① #define N 7
② int i,j,temp,a[N];
③ for (i=0;i<N;i++)
④   scanf("%d",&a[i]);
⑤ for (i=0;i<N/2;i++)
⑥ {
⑦   j=N-1-i;
⑧   temp=a[i];
⑨   a[i]=a[j];
⑩  a[j]=temp;
⑪ }
⑫ for(i=0;i<N;i++)
⑬  printf("%d",a[i]);
运行时输入：
1 2 3 4 5 6 7<回车>
```

第3-4行	依次输入a数组元素的值， a[0]=1,a[1]=2,a[2]=3,a[3]=4,a[4]=5,a[5]=6,a[6]=7
第一次循环	i=0，i<N/2即i<3成立，进入for循环，j=N-1-i=6，三变量法交换a[0]和a[6]的值，因此a[0]更新为6，a[6]更新为1
第二次循环	i=1，i<N/2即i<3成立，进入for循环，j=N-1-i=5，三变量法交换a[1]和a[5]的值，因此a[0]更新为5，a[6]更新为2
第三次循环	i=2，i<N/2即i<3成立，进入for循环，j=N-1-i=4，三变量法交换a[2]和a[4]的值，因此a[0]更新为4，a[6]更新为3
第四次循环	i=3，i<N/2即i<3不成立，循环结束
第12-13行	依次输出a[0]到a[6]的值，即 <b>7654321</b>



# 看代码，写结果 (9)

```
1  #include <stdio.h>
2  #include <string.h>
3  int main()
4  {
5      char s[]="wi\n\x86\0CPU";
6      char a[]="RTX4070i", b[]="Radeon R7";
7      printf("%d\t%d\n",strlen(s), strcmp(a,b)>0);
8      strncpy(a,b,7);
9      puts(a);
10     puts(b);
11
12     return 0;
13 }
```

```
g++ 10  [checkmark] -w
Execution arguments...
Execution stdin...
Program returned: 0
Program stdout
4          0
Radeon i
Radeon R7
```



# 看代码，写结果（9）

- ① char s[]="wi\n\x86\0CPU";
- ② char a[]="RTX4070i", b[]="Radeon R7";
- ③ printf("%d\t%d\n",strlen(s), strcmp(a,b)>0);
- ④ strncpy(a,b,7);
- ⑤ puts(a);
- ⑥ puts(b);

第三行	字符数组s中，s[0]='w',s[1]='i',s[2]='\n',s[3]='\x86',s[4]='\0',s[5]='C',s[6]='P',s[7]='U'; strlen(s)统计的是字符串结束标记'\0'之前的字符数量，因此返回结果4； 比较a和b两个字符串，第一个字符均为'R'，第二个字符'T'<'a'，因此strcmp(a,b)返回值小于0，条件表达式不成立，结果返回0；综上，此行输出 4<若干空格>0
第四行	Strncpy(a,b,7)将b数组的前7个字符拷贝到a数组中，覆盖相应的位置，因此a数组更新为"Radeon i"，b数组不变
第五行	输出 Radeon i
第六行	输出 Radeon R7



# 看代码，写结果 (10)

```
1  #include <stdio.h>
2  #include <string.h>
3  int main()
4  {
5      char s[][10]= {"1234", "56789"};
6      int i,j,sum[2]= {0};
7      for (i=1,j=0; j<2; j++,i=j)
8      {
9          for(; s[j][i]; i*=2)
10             sum[j]=10*sum[j]
11             +s[j][i]-'0';
12         printf("%d %d\n",sum[j],i);
13     }
14
15     return 0;
16 }
```

g++ 10



Compiler options...

Execution arguments...

Execution stdin...

Program returned: 0

Program stdout

23 4

679 8



# 看代码，写结果（10）解析

```
① char s[][10]= {"1234", "56789"};
② int i,j,sum[2]= {0};
③ for (i=1,j=0; j<2; j++,i=j)
④ {
⑤     for(; s[j][i]; i*=2)
⑥         sum[j]=10*sum[j]+s[j][i]-'0';
⑦     printf("%d %d\n",sum[j],i);
⑧ }
```

初始	S数组的状态	<table><tr><td>'1'</td><td>'2'</td><td>'3'</td><td>'4'</td><td>'\0'</td><td>'\0'</td><td>'\0'</td><td>'\0'</td><td>'\0'</td><td>'\0'</td></tr><tr><td>'5'</td><td>'6'</td><td>'7'</td><td>'8'</td><td>'9'</td><td>'\0'</td><td>'\0'</td><td>'\0'</td><td>'\0'</td><td>'\0'</td></tr></table>	'1'	'2'	'3'	'4'	'\0'	'\0'	'\0'	'\0'	'\0'	'\0'	'5'	'6'	'7'	'8'	'9'	'\0'	'\0'	'\0'	'\0'	'\0'
'1'	'2'	'3'	'4'	'\0'	'\0'	'\0'	'\0'	'\0'	'\0'													
'5'	'6'	'7'	'8'	'9'	'\0'	'\0'	'\0'	'\0'	'\0'													
第一次外循环	<p>i=1, j=0, j&lt;2成立，进入循环：</p> <ul style="list-style-type: none"><li>第一次内循环：j=0,i=1,s[0][1]的值是'2'不为0，进入循环，sum[0]=10*sum[0]+s[0][1]-'0' =10*0+'2'-'0'=2，i=i*2=2;</li><li>第二次内循环：j=0,i=2,s[0][2]的值是'3'不为0，进入循环，sum[0]=10*sum[0]+s[0][2]-'0' =10*2+'3'-'0'=23，i=i*2=4;</li><li>第三次内循环，j=0,i=4,s[0][4]的值是'\0'即0，内循环结束；</li><li>输出sum[0]和i的值，即 23 4</li><li>j=j+1=1, i=j=1</li></ul>																					
第二次外循环	<p>i=1, j=1, j&lt;2成立，进入循环：</p> <ul style="list-style-type: none"><li>第一次内循环：j=1,i=1,s[1][1]的值是'6'不为0，进入循环，sum[1]=10*sum[1]+s[1][1]-'0' =10*0+'6'-'0'=6，i=i*2=2;</li><li>第二次内循环：j=1,i=2,s[1][2]的值是'7'不为0，进入循环，sum[1]=10*sum[1]+s[1][2]-'0' =10*6+'7'-'0'=67，i=i*2=4;</li><li>第三次内循环：j=1,i=4,s[1][4]的值是'9'不为0，进入循环，sum[1]=10*sum[1]+s[1][4]-'0' =10*67+'9'-'0'=679，i=i*2=8;</li><li>第四次内循环，j=1,i=8,s[1][8]的值是'\0'即0，内循环结束；</li><li>输出sum[1]和i的值，即 679 8</li><li>j=j+1=2, i=j=2</li></ul>																					
第三次外循环	<p>i=2, j=2, j&lt;2不成立，循环结束</p>																					



# 改错题

先输入100个整数，存入数组a中；再依次从数组a中取出一个整数，如果该整数连续大于该整数之前的五个整数且该整数是偶数（若该数整之前不足五个整数，则不统计），统计出数组a中满足此条件的整数个数cnt并把这些整数按从大到小的顺序存入数组b中；最后，依次输出数组b中的所有整数。

程序第15行之前存在6个错误，其余部分存在若干个错误，请指出程序中的错误并更正。

```
#include <stdio.h> /*第 1 行*/
#define MAX 100; /*第 2 行*/
int main() /*第 3 行*/
{ /*第 4 行*/
    int a[MAX],b[MAX],cnt; /*第 5 行*/
    int i,j,flag; /*第 6 行*/
    for (i=0;i<MAX;i++) /*第 7 行*/
        scanf("%d",a[i]); /*第 8 行*/
    for(i=5;i<MAX;i++)
    {
        for (j=i-4;j<i;j++) /*第 9 行*/
            if (a[i]>a[j]&a[i]%2==0) flag=1; /*第 10 行*/
        else /*第 11 行*/
        { /*第 12 行*/
            flag=0; /*第 13 行*/
            continue; /*第 14 行*/
        } /*第 15 行*/
        if (flag=1) b[cnt++]=a[i]; /*第 16 行*/
    } /*第 17 行*/
    for (i=0;i<cnt-1;i++) /*第 18 行*/
        for (j=i+1;j<cnt-1;j++) /*第 19 行*/
            if (b[i]>b[j]) /*第 20 行*/
            { /*第 21 行*/
                b[i]=b[j]; /*第 22 行*/
                flag=b[i]; /*第 23 行*/
                b[j]=flag; /*第 24 行*/
            } /*第 25 行*/
    printf("cnt=%d\n",cnt); /*第 26 行*/
    for(i=0;i<cnt;i++) /*第 27 行*/
        printf("%6d",b[i]); /*第 28 行*/
    printf("\n"); /*第 29 行*/
    return 0; /*第 30 行*/
} /*第 31 行*/
```

序号	行号	改正
1	第2行	max 后多一个分号，删去
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22-23行	第22行与23行对调
11	第27行	去掉for括号后的;





# 程序设计题 (1)

(12分) 定义“特别数”为一个小于 $2^{31}$ 的整数，且满足其十进制表示含有数字8，或各数位之和为8。编写程序，输入整数 $M$ ， $N$  ( $M < N$ )，将 $M$ 到 $N$ 内（包含 $M$ 和 $N$ ）所有“特别数”输出。

示例输入： 示例输出：

88 120

88 89 98 107 108 116 118

评分标准：（相同功能用其它形式实现的，不扣分）

- (1) 未正确包含头文件扣1分，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，扣1分；
- (3) 未正确书写`for`循环，如`<=end`写为`<end`，扣1分；
- (4) 未正确提取数字，未提取单个位的数字，扣1分；
- (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），扣1分；
- (6) 直接在`i`上修改导致运算后`i`丢失的，扣1分；
- (7) 未正确判断数字8，含判断正确未跳出循环，扣1分；
- (8) 未正确判断各位数字和为8，扣1分；
- (9) 未正确判断输出条件，如`||`写成`&&`，扣1分；
- (10) 未正确书写`main`函数，或写错返回语句，各扣1分；
- (11) 声明语句错，如`start, end, i`等精度小于`int`（可`long long, long double`），每处扣1分，最多扣2分。

```
1  #include <stdio.h>
2
3  int main() {
4      int start, end;
5      scanf("%d %d", &start, &end);    // 输入
6      for (int i = start; i <= end; i++) {    // 遍历范围
7          int sum = 0, n = i, digit = 0;    // n=i避免运算后输出时i丢失
8          while (n) {    // 提取每个位上的数字
9              digit = n % 10;    // 提取单个位上的数字
10             if (digit == 8)    // 判断数字8
11                 break;
12             sum += digit;
13             n /= 10;    // 提取单个位上的数字
14         }
15         if (digit == 8 || sum == 8)    // 判断数字8或者各位数和为8
16             printf("%d ", i);
17     }
18     printf("\n");
19     return 0;
20 }
```





# 程序设计题 (2)

(14分) 现有2个长度不超过100且只包含大小写字母的字符串A和B, 需要统计字符串A中出现字符串B的次数, 不区分大小写。编写程序, 输入2个字符串, 实现上述功能。在"xxx"中查找"xx"记为2次。

要求: 本题不允许调用string.h中的函数。

示例输入:

示例输出:

Thisisadocument IS

2

评分标准: (相同功能用其它形式实现的, 不扣分)

- (1) 整体上, 输入2分, 大小写无关4分, 判断B在A中的功能, 整体6分, 输出2分, 视功能缺失的程度扣分, 不倒扣至负分;
- (2) 未字符串A,B进行大小写无关的比较, 直接比较的, 扣4分; 转换时同转为大写或者同转为小写, 都正确; 转换时先对字符加'A' (或'a') 导致溢出的, 扣2分;
- (3) 使用string.h, 或者strlen, strcmp, strstr, substr, strcat等函数的, 每涉及1个函数扣2分, 最多扣8分;
- (4) 无法判断遍历时A已结束但B未结束的情况, 即aaais中找saa的个数结果为1 (即只判断了字母s就判定成功) 的, 扣3分;
- (5) 循环条件a[j] != '\0' && b[k] != '\0', &&写为||或遗漏条件的, 扣3分; 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未在新一轮循环时初始化为1, 各扣2分;
- (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), 每处错误各扣1分。
- (8) 未能正确使用#include指令, main函数等, 或其它小的编译错误, 每处错误扣1分, 最多扣4分。

```
1  #include <stdio.h>
2  int main() {
3      char a[101], b[101];           // 字符串有100个字符且串尾为0, 长至少101
4      scanf("%s%s", a, b);           // 输入AB, 不取地址
5      int i, j, k, occurrence = 0;    // 记录出现次数应初始化为0
6      for (i = 0; a[i] != '\0'; i++) { // 将AB的同置为大小
7          if (a[i] >= 'a')
8              a[i] -= 'a' - 'A';       // 先求差, 再转换, 避免溢出
9      }
10     for (i = 0; b[i] != '\0'; i++) {
11         if (b[i] >= 'a')
12             b[i] -= 'a' - 'A';
13     }
14     for (i = 0; a[i] != '\0'; i++) {
15         int isfound = 1;             // 假设已找到, 发现不匹配即置为未找到
16         // 在A当中查找B, 在A或B之一达到串尾即退出循环
17         for (j = i, k = 0; a[j] != '\0' && b[k] != '\0'; j++, k++) {
18             if (a[j] != b[k]) {       // 一旦出现字母不匹配, 判定失败
19                 isfound = 0;
20                 break;                // 判断失败退出循环, 未退出也可以
21             }
22         }
23         // 退出是因为找到差异后break, 还是没找完字符串A已达串尾
24         if (isfound == 1 && b[k] == '\0') { // b[k] == '\0' 确保字符串B完整遍历
25             occurrence++;
26         }
27     }
28     printf("%d\n", occurrence);
29     return 0;
30 }
```



# 程序设计题 (3)

(14分) 设有一排灯分别编号1,2,3,4.....,N,  $N \leq 2000$ , 每只灯有亮和暗2种状态。每只灯对应一个开关, 每按一次开关, 对应的灯如果亮着, 则熄灭, 如果暗着, 则点亮。假设刚开始所有的灯都是暗的, 随后, 用户下达指令(a,t), 意思为, 按下编号为{[a],[2a],..., [ta]}且编号不超过N的开关([x]表示 $x$ 向下取整)。编写程序, 输入整数N(灯的总数)和m(指令数), 随后m行为下达的指令, 每行为实数 $a_i$ (小数部分不超过3位)和整数 $t_i$ , 执行完m行指令后, 按从小到大的顺序输出最后亮着灯的编号。

示例输入:                      示例输出:

10 2                      4 5 6 8  
0.32 7  
1.34 6

评分标准: (相同功能用其它形式实现的, 不扣分)

- (1) 整体上, 输入4分, 开关灯功能7分, 输出3分;
- (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, 每处错误扣1分, 最多扣4分;
- (3) 输出语句, 判断灯亮错误的, 或循环的条件错误的(如 $\leq m$ 写为 $< m$ 的), 格式说明符错的, 各扣1分。
- (4) 开关灯功能循环的条件错误的(如 $\leq t$ 写为 $< t$ 的), 每处扣1分。
- (5) 未正确切换亮灯状态的, 扣4分; 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), 每处扣1分, 最多扣3分;
- (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, 每处错误扣1分, 最多扣4分。

```
1  #include <stdio.h>
2  int main() {
3      int s[2001] = { 0 }, m, n;           // 本例使用1..2000故应为2001
4      scanf("%d %d", &m, &n);             // 接受输入灯数和以下行数
5      for (int i = 1; i <= n; i++) {       // 遍历接受每一行输入
6          float a;
7          int t;
8          scanf("%f %d", &a, &t);         // 接受输入a和t
9          for (int j = 1; j <= t; ++j) {    // 遍历1到t用于计算灯号
10             int pos = (int)(j * a);       // 向下取整得到灯号
11             if (s[pos] == 0)              // 切换亮灯状态, 可以有其它等价写法
12                 s[pos] = 1;              // 写为s[pos]=(s[pos]==0)
13             else                          // 或者s[pos]=1-s[pos] 也可以
14                 s[pos] = 0;
15         }
16     }
17     for (int i = 1; i <= m; i++) {        // 依次遍历灯状态, 输出亮灯号
18         if (s[i] == 1) {                  // 判断灯亮
19             printf("%d ", i);
20         }
21     }
22     printf("\n");
23     return 0;
24 }
```



# 你理解递归了吗？

```
1  #include <stdio.h>
2  void hanoi(int n, char c1, char c2, char c3)
3  {
4      if (n==1)
5      {
6          printf("%c->%c\n", c1, c3);
7      }
8      else
9      {
10         hanoi(n-1, c1, c3, c2);
11         printf("%c->%c\n", c1, c3);
12         hanoi(n-1, c2, c1, c3);
13     }
14 }
15 int main()
16 {
17     int n;
18     scanf("%d", &n);
19     hanoi(n, 'A', 'B', 'C');
20     return 0;
21 }
```

人理解循环，  
神理解递归？



吓得我抱起了

抱着抱着抱着我的小鲤鱼的我的我的我

