

第二次练习赛题解

A. HugeGun学姐的圆

解题思路

本题考查点为条件语句，思路比较简单，根据圆的特性进行求解即可，不再赘述。

本题易错点：

1. 如果你使用的是长度的平方，那么你可以直接用 *int* 类型进行数据存储和大小比较，不用考虑误差问题；但如果你使用的是长度，那么你应该用 *double* 类型存的长度，在判断相等与否的时候，要考虑浮点数的误差，比如判断 a, b 是否相等要用 $fabs(a - b) < 1e - 9$ 这样。
2. 本题正解是使用 *if...else if...else* 结构，这样在执行一个分支之后，会直接忽略其他分支。其实用6个 *if* 也并无大碍，只要你6个条件不重叠就可以，但是如果某两个条件重叠了，就会出现一次输出两个字母的情况。所以还是希望大家做题时候细心一点。

AC代码

```
1  #include <stdio.h>
2
3  int main() {
4      int T;
5      int x1, y1, r1, x2, y2, r2, d2;
6      scanf("%d", &T);
7      while (T--) {
8          scanf("%d%d%d", &x1, &y1, &r1);
9          scanf("%d%d%d", &x2, &y2, &r2);
10         d2 = (x1 - x2) * (x1 - x2) + (y1 - y2) * (y1 - y2);
11         if (x1 == x2 && y1 == y2 && r1 == r2) {
12             printf("D\n");
13         }
14         else if (d2 == (r1 + r2) * (r1 + r2)) {
15             printf("Q\n");
16         }
17         else if (d2 == (r1 - r2) * (r1 - r2)) {
18             printf("W\n");
19         }
20         else if (d2 < (r1 + r2) * (r1 + r2) && d2 > (r1 - r2) * (r1 - r2))
21     {
22         printf("E\n");
23     }
24     else if (d2 > (r1 + r2) * (r1 + r2)) {
25         printf("R\n");
26     }
27     else {
28         printf("F\n");
29     }
30     return 0;
31 }
```

B.格式化空格

解题思路

本题考查字符读入，由于只需要对空格进行处理，与换行无关，所以最好的办法就是**逐个字符读入**，如果不是空格就不予处理；如果是空格就把后面的空格全部"吃掉"然后再输出指定个数的空格即可。

AC代码

```
1  #include <stdio.h>
2  int main(){
3      int x;
4      scanf("%d\n",&x);
5      char c;
6      while((c=getchar())!=EOF){
7          if (c!=' '){
8              putchar(c);
9          }
10         else {
11             for (int i=0;i<x;i++)
12                 putchar(' ');
13             do{
14                 c=getchar();
15             }while(c==' ');
16             if (c==EOF)break;
17             else putchar(c);
18         }
19     }
20     return 0;
21 }
22
```

C.花儿街之狼

解题思路

本题一个比较朴素的思路就是利用**二重循环**。也就是把每天的股票价格读入后，对于第*i*天买入的情况，计算后面第*j*天卖出所得收益的极大值，然后再把这*i*个极大值进行比较即可得到答案，这种做法的时间复杂度为 $O(n^2)$ 。

显然本题还有更好的解法。大致思路如下：每读入一天的股票价格，就更新价格的最小值和差的最大值，这样的时间复杂度为 $O(n)$ ，显然快了很多。(其实这里利用了**动态规划**的思想)

关于[时间复杂度](#)的概念，感兴趣的同学可以上网了解一下。

AC代码

这里只提供第一种思路的代码，第二种做法请大家自行尝试。(其实十分简短)

```
1  #include <stdio.h>
2
3  int main(){
4      int n;
5      int a[1005] = {0};
6      scanf("%d",&n);
7      for (int i=0;i<n;i++){
8          scanf("%d",&a[i]);
9      }
10     int max[1005]={0};
11     for (int i=0;i<n;i++){
12         for (int j = i+1;j<n;j++){
13             if(a[j]-a[i]>max[i])
14                 max[i] = a[j] - a[i];
15         }
16     }
17     int maxn = 0;
18     for (int i = 0;i<n;i++){
19         if (maxn<max[i])
20             maxn = max[i];
21     }
22     printf("%d\n",maxn);
23 }
```

D. 字符画拼图

解题思路

首先考试和上机是不会出现这样的题目的，请大家放心。

这题主要考察大家对于数据的存储方式，也就是多维数组的使用，往往好的数据存储和组织方式可以使代码变得更加简洁易读（这一点相信同学们在别的题目中一定也有所体会）。此外就是要善于使用IDE提供的一些基本功能，包括但不限于**查找/替换，调试**等。

AC代码

[illegible]

[illegible]

[illegible]

```

99  "                                     " ,
100 "                                     " ,
101 "                                     " ,
102 "                                     " ,
103 "                                     " ,
104 "                                     " ,
105 "                                     " ,
106 "                                     " ,
107 "                                     " ,
108 "                                     " ,
109 "                                     " ,
110 "                                     " ,
111 "                                     " ,
112 "                                     " ,
113 "                                     " ,
114 "                                     " ,
115 "                                     " ,
116 "                                     " ,
117 "                                     " ,
118 "                                     " ,
119 "                                     " ,
120 "                                     " ,
121 "                                     " ,
122 "                                     " ,
123 "                                     " ,
124 "                                     " ,
125 "                                     " ,
126 "                                     " ,
127 "                                     " ,
128 };
129 int a,b,c,d,i;
130 scanf("%d %d %d %d",&a,&b,&c,&d);
131 for(i=0;i<30;i++)
132     printf("%s%s\n",z[a-1][i],z[b-1][i]);
133 for(i=0;i<30;i++)
134     printf("%s%s\n",z[c-1][i],z[d-1][i]);
135 if(a!=3&&b!=3&&c!=3&&d!=3) printf("Have you finished your homework?");
136 return 0;
137 }

```

E.格式化表格

解题思路

本题主要考察字符串读写，由于涉及到空格且列数未知，所以传统的scanf()方法在此难以奏效，因此比较推荐的一种读写方法是fgets(char, int, FILE),由于不涉及文件读写，所以这里的文件指针只需默认为stdin(标准输入，也就是键盘)即可。

大致解题思路如下：首先初始化所有列类型为“文字”；每读入一行字符，逐个字符进行判断：如果是空格，则转到下一列；否则根据读入的字符和当前列的原类型更新类型，更新的基本原则就是**类型的优先级只降不升**。

AC代码

```
1 //这位同学的代码思路很清晰，所以助教就把它拿过来供大家参考啦
2 /*
3 Author: 陈智勇(34739)
4 Result: AC Submission_id: 1930854
5 Created at: Sat Oct 19 2019 13:51:28 GMT+0800 (CST)
6 Problem: 2557 Time: 5 Memory: 1416
7 */
8 #include <stdio.h>
9 #include <ctype.h>
10 char row[101];
11 char type[101];
12 int main() {
13     while (fgets(row, 101, stdin))
14     {
15         int col = 0;
16         for (int i = 0; row[i] && row[i] != '\r' && row[i] != '\n'; i++)
17         {
18             if (row[i] == ' ')
19             {
20                 col++;
21                 continue;
22             }
23             if (type[col] == '\0')
24             {
25                 type[col] = 'd';
26             }
27             if (row[i] == '.' && type[col] == 'd')
28             {
29                 type[col] = 'f';
30             }
31             if (row[i] != '.' && !isdigit(row[i]))
32             {
33                 type[col] = 's';
34             }
35         }
36     }
37     printf("%s", type);
38     return 0;
39 }
```


F.dlroW desreveR

解题思路

因为需要翻转的字符串中含有空格，所以不能使用 `scanf("%s")` 来输入。可以采用逐个字符输入，然后存到一个字符数组里，用 `-1(EOF)` 来判断输入结束。最后倒着输出即可。

AC代码

```
1  #include<stdio.h>
2  char s[1010];
3  int main()
4  {
5      int i=0;
6      while(1)
7      {
8          char c=getchar();
9          if(c== -1)break;
10         s[i++]=c;
11     }
12     while(i>0)
13     {
14         i--;
15         putchar(s[i]);
16     }
17     return 0;
18 }
```

G.lyd操纵电梯

解题思路

因为 `int` 能表示的范围为 $[-2^{31}, 2^{31})$ 而输入数据范围为 $[-2^{31}, 2^{31}]$ ，而且存在负数，所以推荐使用 `long long` 数据类型，它的表示范围能达到 $[-2^{63}, 2^{63})$ 。

值得注意的是，楼层没有0层，所以当跨越 "0" 楼层时，需要考虑"多"跑一层，比如在 1 层下降 2 层会到达 -2 层而不是 -1 层。

AC代码

```
1  #include<stdio.h>
2  long long l,r,now;
3  int n;
4  int main()
5  {
6      scanf("%lld%lld%lld",&l,&r,&now);
7      scanf("%d",&n);
8      int i;
9      for(i=1;i<=n;i++)
10     {
11         long long x;
12         scanf("%lld",&x);
13         if(now>0)
14             if(x<0&&x+now<=0)now=now+x-1;
15             else now+=x;
16         else
17             if(x>0&&x+now>=0)now=now+x+1;
18             else now+=x;
19         if(now<l)now=l;
20         if(now>r)now=r;
21         printf("%lld ",now);
22     }
23     return 0;
24 }
```

H.来个质数（基础版）

解题思路

质数不能被拆成两个大于1的数的乘积。我们需要判断 n 是否能被而这两个数必一大一小或相等，所以必有一个数小于等于 \sqrt{n} ，所以只需要在 $[2, \sqrt{n}]$ 里寻找能整除 n 的数即可。

注意需要特判 $n = 1$ 的特殊情况

AC代码

```
1  #include<stdio.h>
2  int n;
3  int main()
4  {
5      scanf("%d",&n);
6      int i;
7      int is=1;
8      for(i=2;i*i<=n;i++)
9          if(n%i==0)is=0;
10     if(is==0||n==1)printf("%d",n);
11     else printf("%d",n/2);
12     return 0;
13 }
```

I.进制转化

解题思路

读入完 A 和 B 之后，使用 while 循环读到第一个属于 c 的字符然后连续读完 c。同时根据输入的进制 A 将其转换为10进制数存到变量n里。然后再不断将n对 B 取模得到反过来的 B 进制数的每一位上的数。最后反过来输出，注意大于9的数需要用大写字母表示。

AC代码

```
1  #include<stdio.h>
2  int a,b;
3  int n;
4  char s[35];
5  int main()
6  {
7      scanf("%d%d",&a,&b);
8      char c;
9      while((c=getchar())<33);
10     while(1)
11     {
12         if(c>='0'&&c<='9')
13             n=n*a+c-'0';
14         else if(c>='A'&&c<='K')
15             n=n*a+c-'A'+10;
16         else break;
17         c=getchar();
18     }
19     int tot=0;
20     while(n)
21     {
22         s[tot++]=n%b;
23         n/=b;
24     }
25     int i;
26     for(i=tot-1;i>=0;i--)
27         if(s[i]<10)putchar(s[i]+'0');
28         else putchar(s[i]-10+'A');
29     return 0;
30 }
```

J.毛毛虫和喜鹊

解题思路

使用数组 `vis[10010]` 来记录每个毛毛虫是否已经被吃掉，初始为0表示都没被吃掉。当喜鹊想吃第 i 个毛毛虫时，需要先判断 `vis[i]` 是否为0，吃掉它时将 `vis[i]` 赋值为1表示它被吃掉了。

注意单个毛毛虫长度最大为 10^9 ，那么喜鹊一次吃掉的长度和最大为 $10^{11} > 2^{31} - 1$ ，所以应该使用 `long long` 存答案。

AC代码

```
1  #include<stdio.h>
2  int n,a[10010],m;
3  int vis[10010];
4  int main()
5  {
6      scanf("%d",&n);
7      int i;
8      for(i=1;i<=n;i++)scanf("%d",&a[i]);
9      scanf("%d",&m);
10     for(i=1;i<=m;i++)
11     {
12         int k;
13         long long ans=0;
14         scanf("%d",&k);
15         while(k--)
16         {
17             int x;scanf("%d",&x);
18             if(!vis[x])ans+=a[x],vis[x]=1;
19         }
20         printf("%lld\n",ans);
21     }
22     return 0;
23 }
```

K. 花式阶乘

解题思路

本题考查的是数组的应用。

先看数据范围， $0 < n \leq 10000$ ，用头发想想都能想到结果肯定很长，怎么存那么大的数呢？答案是用数组。

我们可以模拟手算的时候列式计算的过程，把每个数依次地乘进去，最后进行格式处理再输出就行。

btw，做这种题的时候一定要细心，弄清楚你每一步要干什么，你数组里的数字是怎么存的，这些都清楚了之后，你才能A掉这道题。

剩下的就直接看代码吧。

AC代码

```
1  #include <stdio.h>
2
3  int main() {
4      int num[50001] = {1}; //低位存数字的低位，高位存数字的高位，初始化最低位是1，高位都是0
5      int i, j, n, bottom = 0, len = 1, overflow = 0, tmp;
6      //len存一下现在数字有多长，overflow表示进位，tmp是临时变量，bottom在最后标记最低位的非0数字
7      scanf("%d", &n);
8      for (i = 1; i <= n; i++) { //遍历1-n
9          for (j = 0; j < len; j++) { //将现在的数字每一位乘上i，模拟列式计算的过程，注意处理进位
10              tmp = num[j] * i + overflow;
11              overflow = tmp / 10;
12              num[j] = tmp % 10;
13          }
14          while (overflow != 0) { //乘完了还有进位，那就接着往后存
15              num[len] = overflow % 10;
16              overflow /= 10;
17              len++;
18          }
19      }
20      while (num[bottom] == 0) { //将低位的0忽略掉
21          bottom++;
22      }
23      for (i = len - 1; i >= bottom; i--) { //从高位到低位依次输出，输出第二个数字之前加个小数点
24          if (i == len - 2) {
25              printf(".");
26          }
27          printf("%d", num[i]);
28      }
29      printf("*10^%d", len - 1); //把指数部分输出了
30      return 0;
31 }
```