第四次上机题解

写在前面

这次上机题目没有太难，也没太简单，属于正常水平吧。上机过程中发现这么几点问题，希望大家注意一下。

1. B题，求pai，有的同学没有使用pow函数，而是自己推导的结论，这是没有问题的，但问题就出在使用变量的时候要注意数据范围，不要用int型整数去存239^{2i+1}，这肯定是会爆数据范围的，对于本题要用$double$才可以。
2. 字符串处理有关的题一定要细心！一定要细心！一定要细心！
3. 定义了什么类型的变量，就要用什么类型的$scanf$，有的同学定义了$long\ long\ a$，却是使用$\%d$读入的，这肯定是会出问题的。

A. 名次预测

解题思路

本题读入名次信息后，可以使用 for 循环判断名次预测是否准确，也可以用AC代码的方法判断。

对于形如 $a == 1$ 的表达式，若表达式成立则为1，否则为0。

AC代码

```
#include<stdio.h>
int main()
{
    int s[6],i;
    while(scanf("%d %d %d %d %d %d",&s[0],&s[1],&s[2],&s[3],&s[4],&s[5])!=EOF)
    {
        i=(s[0]==1)+(s[1]==2)+(s[2]==3)+(s[3]==4)+(s[4]==5)+(s[5]==6);
        if(i==3||i==4) printf("Yes\n");
        else printf("No\n");
    }
    return 0;
}
```

B. Ganten的圆周率

解题思路

本题读入 n 后使用 for 循环和 $math.h$ 库中的 pow 函数计算即可。

AC代码

```
#include<stdio.h>
#include<math.h>
int main()
{
    int n,i;
    scanf("%d",&n);
    double ans=0,a,b;
    for(i=0;i<n;i++)
    {
        a=16*pow(-1,i)/((2*i+1)*pow(5,2*i+1));
        b=4*pow(-1,i)/((2*i+1)*pow(239,2*i+1));
        ans=ans+a-b;
    }
    printf("%.12lf",ans);
    return 0;
}
```

C. switch优化

解题思路

本题按照题目要求，将 $if-else$ 语句替换为 $switch-case$ 语句即可。请注意 $case$ 中需要视情况添加 $break$ 语句。

AC代码

```
#include<stdio.h>
int main(){
int a, b, c, x, y;
int ans = 0;
int t = 0, i;
char op;
scanf("%d%d%d", &t, &x, &y);
for(i = 1; i <= t; i++){
    op = ((x * i)  + y)&15;
    a = i ^ 987654321;
    b = i ^ 123456789;
    switch(op)
    {
        case 0: ans += a + b; break;
        case 1: ans += a - b; break;
        case 2: ans += a * b; break;
        case 3: ans += a / b; break;
        case 4: ans += a % b; break;
        case 5: ans += a & b; break;
        case 6: ans += a | b; break;
        case 7: ans += a ^ b; break;
        case 8: ans -= a + b; break;
        case 9: ans -= a - b; break;
        case 10: ans -= a * b; break;
        case 11: ans -= a / b; break;
        case 12: ans -= a % b; break;
        case 13: ans -= a & b; break;
        case 14: ans -= a | b; break;
        case 15: ans -= a ^ b; break;
    }
}
printf("%d", ans);
}
```

D. 表达式求值（简单版）

解题思路

本题逻辑十分清晰，每个表达式先读入第一个数字，之后每次读入一个运算符和一个数字，并跟前面的值进行运算即可。

本题要注意的几个问题，其实在hint里都提到了：

1. 针对字符读入问题，可以通过在格式串中增加空格的方式忽略空格，如：使用`scanf(" %c%d", &ch, &n)`这样可以读入`+ 2`这样的输入，确保读入后ch为+，n为2。
2. 请注意数据的取值范围是$2^{63}-1$，所以要用 $long\ long$ 来存储数据。

AC代码

```
#include <stdio.h>

int main() {
    long long n, sum, number;
    char op;
    scanf("%lld", &n);
    scanf("%lld", &sum);
    while (n--) {
        scanf(" %c", &op);
        scanf("%lld", &number);
        if (op == '+') {
            sum = sum + number;
        }
        else if (op == '-') {
            sum = sum - number;
        }
        else if (op == '*') {
            sum = sum * number;
        }
        else if (op == '/') {
            sum = sum / number;
        }
    }
    printf("%lld", sum);
    return 0;
}
```

E. 选专业

解题思路

本题考查点是怎么读入n个数据。这种情况我们可以使用数组读入。在第一行读入了n的前提下，在后面使用`for(i=0;i<n;i++) scanf("%lf", &s[i]);`即可读入n个浮点数。在处理完读入的问题后，后面按照公式进行计算即可。

AC代码

```
#include <stdio.h>

int main() {
    int i, n;
    double s[40], a[40], sumGrade = 0, sumGPA = 0, sumScore = 0;
    scanf("%d", &n);
    for (i = 0; i < n; i++) {
        scanf("%lf", &s[i]);
        sumScore += s[i];
    }
    for (i = 0; i < n; i++) {
        scanf("%lf", &a[i]);
        sumGrade += s[i] * a[i];
        if (a[i] >= 60) {
            sumGPA += (s[i] * (4 - (100 - a[i]) * (100 - a[i]) * 3 / 1600));
        }
    }
    printf("%.2f\n%.3f", sumGrade / sumScore, sumGPA / sumScore);
    return 0;
}
```

$Hint$：大家第一学期结束以后可以用你写的这个程序算算自己的成绩：）

F. Comment Extractor

解题思路

题目保证每行输入字符串存在且只存在两个星号和两个斜杠，且是以一个`/*`和一个`*/`的形式出现，且`/*`一定出现在`*/`之前。 那么我们只需要把`/*`和`*/`的位置找到就能解决这个问题。

星际玩家请注意：$NOTHlNG$里面是l不是I。所以请大家记住，在要输出指定的字符串的时候，一定要复制粘贴，一定要复制粘贴，一定要复制粘贴，不要手敲。

AC代码

```
#include<stdio.h>
#include<string.h>
char s[300];
int n;

int main()
{
    while (~scanf("%s", s)) {
        int i, st = 0, en = 0;
        n = strlen(s);
        for (i = 0; i < n; ++i) {
            //寻找 /*
            if (s[i] == '/' && s[i + 1] == '*')
                st = i + 2;
            //寻找 */
            if (s[i] == '*' && s[i + 1] == '/')
                en = i - 1;
        }
        if (st > en) puts("NOTHlNG");
        else {
            s[en + 1] = '\0';
            printf("%s\n", s + st);
        }
    }
    return 0;
}
```

st为要输出的部分的起点，en为要输出部分的终点。

当$st > en$时说明这条注释里面什么都没有，否则输出中间这段即可。

代码里`printf("%s\n", s + st);`表示我们从字符串s的st位置开始输出，当用$printf$输出字符串时它会停在st后的第一个`'\0'`的位置，所以程序中把$s[en+1]$这个位置设置为`'\0'`。平时我们在输出字符串时是相当于`printf("%s", s + 0);`即从头开始输出。大家可以自行尝试。

下面附上出题人tky大佬的代码：

```c
#include <stdio.h>

int main()
{
    char s[666], t[666];
    while (gets(s))
        *t = 0, sscanf(s, "%*[^*]*%[^*]*%*s", t), puts(*t ? t : "NOTHlNG");
}
```

这里面涉及到了$sscanf$和正则表达式的用法，有兴趣的同学可以自行百度，这些不会作为考察点。

当然，不建议用$gets$函数，虽然有时候能过题，但是总有 WA 的时候 (x。

G. 恰当的时间

解题思路

这道题的难点在于时间的计算，我们可以考虑把他们转换为相对于$00:00$过了多少分钟，即$hour*60+minute$

题目保证了起止时间间隔小于$24h$，这说明如果路人C熬夜了，那么有$h_1>h_2$，这时我们可以把h_2加上24，即我们把第二天的x点当作第一天的$24+x$点。

这样我们在计算两个时间的差$delta$的时候就会很方便，即$(h_2-h_1)*60+m_2-m_1$，我们并不需要考虑m_1和m_2的大小，想一想，为什么?

这样我们就能计算出第二项作业开始时间now，我们最后将它转换为小时和分钟。

小时部分需要对24取模，因为我们可能会在第二天开始写第二项作业。

AC代码

```
#include<stdio.h>
int main()
{
    int x,y,h1,m1,h2,m2,h3,m3;
    scanf("%d%d%d:%d%d:%d",&x,&y,&h1,&m1,&h2,&m2);//scanf("%d:%d",&h,&m)过滤`:`
    if(h2<h1)h2+=24;
    int delta=(h2-h1)*60+m2-m1,now=h1*60+m1+delta*x/(x+y);
    printf("%02d:%02d",now/60%24,now%60);
    return 0;
}
```

H. 自动卖菜机

解题思路

一道题目怎么说就怎么做的模拟题。

我们可以使用数组来进行计数，因为种类很少也可以使用if。

需要注意的是钱数只有在投入5和10的时候才会增加。

AC代码

```
#include <stdio.h>
int num[4] = {0},mon = 0,aa,opt,bb;
int main() {
    int n;
    scanf("%d",&n);
    for (int i = 1;i <= n; i++) {
        scanf("%d",&opt);
        if (opt == 1) {
            scanf("%d%d",&aa,&bb);
            num[aa] += bb;
        } else if (opt == 2) {
            scanf("%d",&aa);
            if (aa == 5 || aa == 10) {
                mon += aa;
            }
        } else {
            scanf("%d", &aa);
            if (num[aa] && mon >= 5) {
                num[aa]--;
                if (aa == 1)
                    printf("apple");
                else if (aa == 2)
                    printf("banana");
                else
                    printf("carrot");
                mon -= 5;
                if (mon)
                    printf(" %d",mon);
                printf("\n");
                mon = 0;
            }
        }
    }
    return 0;
}
```

I. 五边形

解题思路

这个题的数据范围很小，我们可以枚举每个人拿出的木棍的长度，最后看他们能否拼成一个五边形。

我们还需要对五边形的周长去重，观察到木棒长度不超过10000，那么五边形周长不会超过50000，那么我们可以使用一个数组来记录某个周长是否出现，这是在第三次上机中出现过的技巧。

那么如何枚举每个人拿出的木棍，这个做法和第三次练习赛中`zyy学姐的生日礼物5`一样，使用多重循环即可。

最后问题就只剩下了如何判断五根木棍能够组成一个五边形，这个问题的本质和判断三根木棍能否组成三角形是一样的，只要 **总长度**$>2*$**最长的木棍的长度** 就能组成一个五边形。

AC代码

```
#include <stdio.h>
#include <string.h>
int flag[50005],dd,ee,ff,gg,hh;
int d[20],e[20],f[20],g[20],h[20];

int max(int a, int b) {
    if (a > b) return a;
    return b;
}

int main()
{
    while(~scanf("%d%d%d%d%d",&dd,&ee,&ff,&gg,&hh))
    {
        int i,j,k,l,m;
        for(i=1;i<=dd;i++) scanf("%d",&d[i]);
        for(i=1;i<=ee;i++) scanf("%d",&e[i]);
        for(i=1;i<=ff;i++) scanf("%d",&f[i]);
        for(i=1;i<=gg;i++) scanf("%d",&g[i]);
        for(i=1;i<=hh;i++) scanf("%d",&h[i]);
        memset(flag,0,sizeof(flag));
        for(i=1;i<=dd;i++)
            for(j=1;j<=ee;j++)
                for(k=1;k<=ff;k++)
                    for(l=1;l<=gg;l++)
                        for(m=1;m<=hh;m++)
                        {
                            int
maxx=max(max(max(d[i],e[j]),max(f[k],g[l])),h[m]);
                            int sum=d[i]+e[j]+f[k]+g[l]+h[m];
                            if(sum-maxx>maxx) flag[sum]=1;
```

```
                        }
        int ans=0;
        for(i=1;i<=50000;i++) ans+=flag[i];
        printf("%d\n",ans);
    }
    return 0;
}
```

为了方便这份代码使用了函数，当然你不会函数也是可以用简单的4个if解决的。

```
int maxx=d[i];
if(e[j]>maxx) maxx=e[j];
if(f[k]>maxx) maxx=f[k];
if(g[l]>maxx) maxx=g[l];
if(h[m]>maxx) maxx=h[m];
```

J. 小糖与小光的知识问答（1）

解题思路

本题考查点是蔡勒公式的应用。蔡勒公式是根据日期求星期的一个公式，比较好用。本题的大致思路是：如果要判断某一天是星期几，则直接使用蔡勒公式进行计算即可；如果要你计算某个月的第n个星期w的日期，则可以用循环结构遍历那个月的所有天，分别用蔡勒公式计算星期，当恰好是第n个星期m的时候，输出当天的日期。

本题的技巧有：

1. 使用函数封装蔡勒公式，便于多次使用（如果你不会用函数也是没关系的，因为还没有讲到，直接计算就可以）；
2. 使用数组存储每个月的天数，以及一周七天的名字。

AC代码

```
#include <stdio.h>

int month[14] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
char week[10][5] = {"Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"};

int is_leap(int y) {
    return (y % 4 == 0 && y % 100 != 0) || y % 400 == 0;
}

int zeller(int y, int m, int d) {
    int c, ans;
    if (m < 3) {
        m += 12;
        y--;
    }
    c = y / 100;
    y = y % 100;
    ans = c / 4 - 2 * c + y + y / 4 + 26 * (m + 1) / 10 + d - 1;
    while (ans < 0) {
        ans += 7;
    }
    return ans % 7;
}

int main() {
    int op, y, m, d, i, n, w, cnt = 0;
    char s[10];
    scanf("%d%s", &op, s);
    y = (s[0] - '0') * 1000 + (s[1] - '0') * 100 + (s[2] - '0') * 10 + (s[3] -
'0');
    m = (s[4] - '0') * 10 + s[5] - '0';
```

```
    month[2] += is_leap(y);
    if (op == 1) {
        d = (s[6] - '0') * 10 + s[7] - '0';
        if (m < 1 || m > 12 || d > month[m] || d < 1) {
            printf("bad question");
        }
        else {
            printf("%s", week[zeller(y, m, d)]);
        }
    }
    else if (op == 2) {
        n = s[6] - '0';
        w = s[7] - '0';
        if (m < 1 || m > 12) {
            printf("bad question");
        }
        else {
            for (i = 1; i <= month[m]; i++) {
                if (zeller(y, m, i) == w) {
                    cnt++;
                }
                if (cnt == n) {
                    printf("%04d%02d%02d", y, m, i);
                    break;
                }
            }
            if (cnt < n) {
                printf("bad question");
            }
        }
    }
    return 0;
}
```

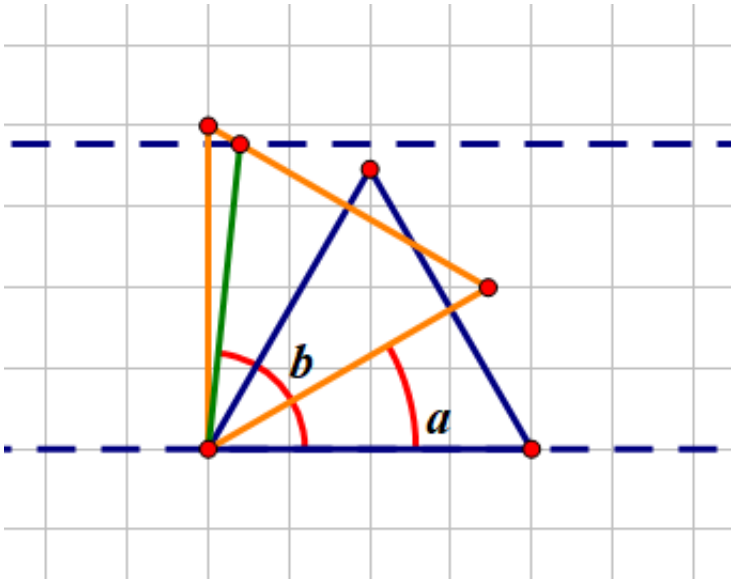
K. HugeGun学姐过生日咯

解题思路

设x轴平行于所有直线，那么显然三角形和直线相交的概率取决于三角形在y轴上投影的长度。因此要对角度进行积分。当投影长度为x时，相交的概率为：

$$p(x)=\begin{cases}x/L & L>x\\ 1 & L\le x\end{cases}$$

根据等边三角形的对称性，我们只需要对下图中绕点旋转的三角形对角度进行积分再求平均。



其中a为需要积分的角度区间$[0,\pi/6]$；b为临界角度，当三角形旋转角度到超过b时p为1，否则p为x/L

若$l<L$则令$b=\pi/2$

答案为

$$\frac{1}{\pi/6}\int_0^{\pi/6}p(l\sin(\theta+\pi/3))d\theta=\frac{1}{\pi/6}(\int_{\pi/3}^{b}\frac{l\sin(\theta)}{L}d\theta+\int_b^{\pi/2}1d\theta)$$

AC代码

```
#include<stdio.h>
#include<math.h>
int n,l,L;
int main()
{
    scanf("%d",&n);
    while(n--)
    {
        scanf("%d%d",&L,&l);
        if(3*l*l>=4*L*L){puts("1.000000");continue;}
        double rad,pi=acos(-1),ans=0;
        if(l>L)rad=asin(1.*L/l);
```

```
        else rad=pi/2;
        ans+=pi/2-rad;
        ans+=(-cos(rad)+cos(pi/3))*l/L;
        ans/=(pi/6);
        printf("%.6f\n",ans);
    }
    return 0;
}
```