A 雪容融

难度	考点
1	特殊字符输出

题目分析

对于这种带有大量特殊字符的字符画，如果一个个手动更换大概率会出错，因此建议先复制到记事本上替换后再粘贴到dev c++上，方法可参考B站上的视频。

空格与换行等容易忽视的字符也要引起注意，否则失之毫厘谬之千里。

参考代码

```
#include<stdio.h>
#include<string.h>
int main(){
    printf("\
                                    \'\'\'\'\'\'\'*,`\'\'\'\'\n\
                                   \'\'\'ooo[`\'\'\',o`\'\'\'\n\
                                    \'=^\'\'\'\'\'   \'\\o`\'\'\n\
                                   \'\'\'=\\*o\\\'\',]`\'\'O^\'\n\
                         \'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\',o^,o/`\'\'\'\n\
                   \'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\n\
             \'\'\'\'\']]/%%OoOOO]`\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\n\
 \'\'\'\'/%%OOO/]O%%O`\'\\%%%%%%O*\'\'\'\'\'\'\'\']]]\'\'\'\'\'\'\'\'\'\'\'\'\'\n\
 \'\'\',/%%%%%%%%%%%%%%%%%%%%\\]OO`\',O%%%%O`=OO%%%%%%OOO%%O]\'\'\'\'\'\'\'\'\'\'\'\'\n\
\'\',%%%%%%%%%%%%%%%%%%%%OOO%%%%%%%%%%%%%%%%%%%%O]]/O^\'\'/%%%%^[%%%%\\\'\'\'\',]\'\'\'\'\'\'\n\
\'\'\'\',%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%OOOOOOO%%%%%%%%%%%O%%%%%%%%\\\'=O/*,/OO%%%%\\\'\'\'\'\'\n\
\'\',O%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%OOOOOOOOOO%%%%%%%%%%%%%%%%%%\\/%%%%%%^,O^/`\'\'\'\'\n\
\'\'\'=%%%%%%%%%%%%OOO%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%OOOOOOOO%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%O%%%%]*\'\'\'\n\
 \'=%%%%%%%%%%OOOO%%%%%%%%%%%%%%%%%%%%%%OOOO%%%%%%%%%%%OOOOOOOO%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%^\'\'\n\
```

```
  \'\',%%%%%%%%%%OOO%%%%%%%%%%%%%%%%%%
[\'\'\'\'\'\'\'\'\'[O[[[O%%%%%%%%%%%%%%%%%%%%OOOOOOoO%%%%%%%%%%%%%%%%%%%%%%%`\'\n\
  \'\'O%%%%%%%%%%OO%%%%%%O[*\'\'\'******\'\'\'\'\'\'\'\'\'\'\'\'\'\\%%%%%%%%%%%%%%OO*=O*\
'*oO%%%%%%%%%%%%%%%%%%%%%`\'\'\n\
  \'=%%%%%%%%%%%%%%%%%%O`*`*************\'\'\'\'\'\'\'\'\'\'\'=%%%%%%%%%%%%%%%%Ooo*\'**\\
O%%%%%%%%%%%%%%%%%%%%%%\'\'\n\
  \'=%%%%%%%%%%%%%%%%%%^,oooo`***********\'\'\'\'\'\'\'\'\'\'=%%%%%%%%%%%%%%%%%%OOOOOOOO%
%%%%%%%%%%%%%%%%%%%%%^\'\n\
  \'O%%%%%%%%%%%%%%%%O\\oooooooo%%%%^**********\'\'\'\'\',%%%%%%%%O[[[O%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%`\'\'\n\
  \'=%%%%%%%%%%%%%%^/oooOooo=%%%%%%O*,**********\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\',%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%^\'\'\'\n\
  \',%%%%%%%%%%%%%%^\\oOOOOOoooO[ooo`************\',`\'\'\'\'\'\'\'\'\'\',%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%^\'\'\'\n\
  \'\'O%%%%%%%%%%%%O/oOOOOOoooooooooo\\/\\********\'%%%%%%O\'\'\'\'\'\'\'\'\'\'O%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%^\'\'\'\n\
  \'\',%%%%%%%%%%%%%%%%\\/ooooooooooooooooo`*****,]O%%%%`\'\'\'\'*****\'O%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%^\'\'\n\

 \'=%%%%%%%%%%%%%%%%%%%%OOO%%OO`oooooooooooooooo`********,oooo^\'=%%%%%%%%%%%%%%%%%%%%
%%%%%O\'\n\
   \'\'\'=%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%O],
[oooooooooo`*********,ooOo/\',%%%%%%%%%%%%%%%%%%%%%%%%%%`\n\

 \',%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%O]]//[[[[[[**,]]O%%%%%%%%\\***,/%%%%%%%%%%%
%%%%%%%%%%%%%%%^\'\'\n\

\'\'\'\\%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%^\'\n\

 \'\',O%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%O`\'\'\n\

 \'\',%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%/\'\'\n\

\'\'\'O%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%/\'\'\'\n\

 \'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%O`\'\'\'\n\

 \'O%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%O[\'\'\'\n\

 \'\'O%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%OOOOOOOOOOOOooOOO`\'\'\'\n\

\'\',%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%OOOOOOOOOO%%%%%%%%%%%%%%%%%%%%O%%O\'\n\
```

```
                 \'[O%%%%%%%%%%%%%%%%%%OO[\'\'\'/\\`\'*
[\\O%%%%%%%%%%%%%%%%%%%%%%%%O%%%%`\'\'\n\
                 \'
\'/%%%%%%%%%%%%O`\'\'\'O%%%OO`\'\'\'*oO%%%%%%%%%%%%%%%%%%%%%%%%%%%%O\'\'\n\

 \'\'=%%%%%%%%%%%%O`\'\'\'\',o`\'\'\'\'\'\'\'*O%%%%%%%%%%%%%%%%%%%%%%%%%O\'\n\

 \'\'=%%%%%%%%%%%%o\'\',]`*\'\'\'\'\'\']\'\'\'\\O%%%%%%%%%%%%%%%%%%%%%%%%%^\'\'\n\
                  \'\'=%%%%%%%%%%%%O*\'\'[\\/[[[,
[[[`\'\'O%%%%%%%%%%%%%%%%%%%%%%/`\'\'\'\n\

 \'\'\'O%%%%%%%%%%Oo\'\'\'\',*\'\'\'\'\'\'\'/O%%%%%%%%%%%%%%/\'\'\'\n\
                     ,%%%%%%%%%%%%%%O]\'\'\'\\**`\'\',O%%%%%%%%%%%%%%%%%^\'\'\n\
                    \'\',%%%%%%%%%%%%%%%%OO\\]]]]/OOOO%%OO%%%%%%%%%O\'\'\'\n\

 \'\'\',%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%OO%%/\'\'\'\'\n\
                      \' ,%%%%%%%%%%%%%%%%%%%%%%%%O%%%%%%%%%%%%%%%%%%%%%%\'\'\'\'\n\
                      \'\'\',%%%%%%%%%%%%%%%%%%%%^\'\\%%%%%%%%%%%%%%%%[\' \'\n\
                          \'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'\'");
    return 0;
}
```

B 时刻1.0

难度	考点
1	判断

题目分析

本题较为简单，使用 if-else 判断即可。

参考代码

```
#include <stdio.h>
int main()
{
    int h1, m1, h2, m2;
    scanf("%d:%d", &h1, &m1);
    scanf("%d:%d", &h2, &m2);
    if (h1 < h2)
        printf("2");
    else if (h1 > h2)
        printf("1");
    else
    {
```

```
        if (m1 < m2)
            printf("2");
        else if (m1 > m2)
            printf("1");
        else
            printf("Same Time");
    }
    return 0;
}
```

C 来计算圣遗物评分吧！

难度	考点
1	简单计算

题目分析

一道很简单的题目，只需要把每一行的数据读入进来，然后按照题目里给的公式将第一个数 `*` 2+第二个数+第三个数 `*` 0.25+第四个数计算出来即可。

很多PE可能是没有换行吧quq

参考代码

```
#include<stdio.h>
int main(){
    double a[4], sum;

    int m=5, i;
    while(m --){
        for(i=0; i<4; ++i)
            scanf("%lf", &a[i]);
        sum = a[0]*2.0 + a[1]*1.0 + a[2]*0.25 + a[3]*1.0;
        printf("%.2lf\n", sum);
    }
    return 0;
}
```

D 猜猜我是倒还是不倒

难度	考点
1	循环

题目分析

需要写两层循环：

第一层：控制有$(n/2)$行（n是int类型，进行除法会自动向下取整）。

第二层：每一行的空格数、“*”数与行数的关系

参考代码

```
#include <stdio.h>
int main()
{
    int n, op;
    scanf("%d%d", &n, &op);
    if (!op)
    {
        for (int i = 0; i < n / 2 + 1; i++)
        {
            for (int j = 0; j < n / 2 - i; j++)
                putchar(' ');
            for (int j = 0; j < 2 * i + 1; j++)
                putchar('*');
            putchar(10);
        }
    }
    else
    {
        for (int i = 0; i < n / 2 + 1; i++)
        {
            for (int j = 0; j < i; j++)
                putchar(' ');
            for (int j = 0; j < n - 2 * i;j++)
                putchar('*');
            putchar(10);
        }
    }
}
```

E 中间位取数

难度	考点
2	循环、逻辑判断

题目分析

C1已经有了对于整数拆位的题目，此题不同的地方是不知道给出的整数位数，所以需要用到循环，给出拆位的通用代码：

```
while(x!=0){
    a[++cnt]=x%10;
    x/=10;
}
```

注意特判x=0的特殊情况。另外用字符数组可以使用strlen判断位数，大家可以自行思考。

参考代码

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

int x, cnt, a[15];

int main()
{
    scanf("%d", &x);
    if (x == 0)
    {
        printf("0");
        return 0;
    }
    while (x != 0)
    {
        a[++cnt] = x % 10;
        x /= 10;
    }
    if ((cnt & 1) == 1)
        printf("%d", a[cnt / 2 + 1]);
    else
        printf("%d", a[cnt / 2] + a[cnt / 2 + 1]);
}
```

F 求最大斜率

难度	考点
2	数组、浮点数运算

问题分析

本题主要考察数组的简单运用、浮点数的输入输出、运算以及大小比较。

通过二重循环的方式枚举所有点对并运用斜率公式计算其斜率，选出其中最大值即为答案。

如果不是很理解二重循环代码的运行方式，可以通过调试一步一步看各语句的执行顺序。

参考代码

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<math.h>
int n;
int x[1010],y[1010]; //设置数组时一般会比题目所给的数据范围大一些，以防越界
int main()
{
    scanf("%d",&n);
    for(int i=1;i<=n;i++)
        scanf("%d%d",&x[i],&y[i]);
    double ans=-2e7; //首先将答案设置为极小值
    for(int i=1;i<=n;i++) //枚举第i个点
    for(int j=i+1;j<=n;j++) //枚举第j个点，考虑不重复枚举所以从i+1开始
    {
        double tmp=1.0*(y[i]-y[j])/(x[i]-x[j]); //计算点i和点j的斜率
        if(tmp>ans) ans=tmp; //与当前答案进行比较，若点i和点j的斜率大于答案则更新答案
    }
    printf("%.3lf\n",ans); //输出答案并保留三位小数
    // system("pause");
    return 0;
}
```

思考一下：当$0 \leq n \leq 10^5$时，这份代码显然会超时，那我们又该怎样解决这个问题呢?

G 驼峰命名与短横线命名

难度	考点
2	简单字符处理

题目分析

本题可以用 `HINT` 中读字符的方法来读取数据，只需要对读入的字符分类即可，请见示例代码。

示例代码

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

int main()
{
    char c;
    while ((c = getchar()) != EOF)
    {
        if (isupper(c))
        { // 读到大写字母（驼峰命名），则输出连字符，并将字母转小写输出
            printf("-%c", tolower(c));
        }
        else if (c == '-')
        { // 读到连字符（短横线命名），则再读一个字符，转为大写输出
            c = getchar();
            printf("%c", toupper(c));
        }
        else
        { // 其他字符照常输出
            printf("%c", c);
        }
    }
    return 0;
}
```

H 计算 ln2

本题很简单。直接看代码：

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

int main()
{
    double d=log(2);
    double sum=0;
    int k;
    scanf("%d",&k);
    int i;
    for(i=1;i<=k;i++)
```

```
    {
        if(i%2==1)
        {
            sum+=1.0/(double)i;
        }
        else
        {
            sum-=1.0/(double)i;
        }
    }
    if(k%2==1)
    {
        printf("up %.6lf",sum-d);
    }
    else
    {
        printf("down %.6lf",d-sum);
    }
}
```

I zhn の 奇妙魔法

难度	考点
3	思路题

题目分析

本题注意题目中给出的可以选择任意个位置的任意个数字，不难发现，最终的最小步数只需要输出最大值减去最小值即可。

注意多组数据输入，每组的最大值与最小值设定需要更新。

参考代码

```
#include <stdio.h>
int i, j, n;
int a[100], t;
int main()
{
    scanf("%d", &t);
    while (t--)
    {
        scanf("%d", &n);
        int mx = 0, mn = 1e9;
        for (j = 1; j <= n; j++)
```

```
        {
            scanf("%d", &a[j]);
            if (a[j] > mx)
                mx = a[j];
            if (a[j] < mn)
                mn = a[j];
        }
        printf("%d\n", mx - mn);
    }
    return 0;
}
```

J 810975

难度	考点
3	多层嵌套循环

题目分析

本题难点主要在于“七鸡”和“五连鸡”的判断。一种方式是，用几个变量分别记录：

- 总鸡数
- 当前连鸡数
- 最大连鸡数

即可判断。

示例代码

```
#include <stdio.h>

int rank[110];

int main()
{
    int n = 0, i = 0, j = 0, index = -1;
    scanf("%d", &n);
    for (i = 0; i < n; i++)
    {
        scanf("%d", &rank[i]);
    }
    if (n >= 9)
    {
        for (i = 8; i < n; i++)
        {
            int chp = 0, cchp = 0, mcchp = 0;
```

```
            for (j = (i - 8); j <= i; j++)
            {
                if (rank[j] == 1)
                {
                    chp++;
                    cchp++;
                }
                else
                {
                    if (mcchp < cchp)
                    {
                        mcchp = cchp;
                    }
                    cchp = 0;
                }
            }
            if (mcchp < cchp)
            {
                mcchp = cchp;
            }
            if (chp == 7 && mcchp == 5)
            {
                index = i + 1;
                break;
            }
        }
    }
    printf("%d", index);
    return 0;
}
```