

# 第三次练习赛题解

--HugeGun

A~D题为第三次上机原题，请参看第三次上机题解。

## E.Zyy学姐的生日礼物

正确率25%

### 解法一：

使用二维数组处理出杨辉三角的前一千行，最后根据输入的行数进行输出。**请注意运算过程中取模。**

```
#include<stdio.h>
const int mod = 1e9 + 7;
int n, i, j;
int ans[1010][1010];

int main()
{
    ans[1][1] = 1;
    for (i = 2; i <= 1000; ++i)
    {
        ans[i][1] = ans[i][i] = 1;
        for (j = 2; j < i; ++j)
            ans[i][j] = (ans[i - 1][j - 1] + ans[i - 1][j]) % mod;
    }

    while (scanf("%d", &n) == 1) {
        for (i = 1; i <= n; ++i)
            printf("%d ", ans[n][i]);
        printf("\n");
    }

    return 0;
}
```

### 解法二(\*有兴趣同学参考):

直接计算杨辉三角对应行的值，即 $n$ 个组合数。

先预处理出阶乘数组，利用公式 $C_n^m = \frac{n!}{m! \times (n-m)!}$ ，计算出 $n$ 个组合数。注意到在取模运算后无法使用除法(存在除不尽的情况)，因此利用我们的模数 $1e9 + 7$ 是一个质数，再预处理出每个阶乘的逆元，用乘逆元代替除法即可。

```

#include<stdio.h>
typedef long long LL;
const LL mod=1e9+7;
LL qpow(LL a,LL b)
{
    LL ret=1;
    while(b)
    {
        if(b&1)ret=ret*a%mod;
        b>>=1;
        a=a*a%mod;
    }
    return ret;
}
LL inv(LL a){return qpow(a,mod-2);}
LL jie[1111],inv_jie[1111];
LL C(int n,int m)
{
    return jie[n]*inv_jie[n-m]%mod*inv_jie[m]%mod;
}
int main()
{
    jie[0]=1;inv_jie[0]=1;
    int i,n;
    for(i=1;i<=1000;i++)jie[i]=(jie[i-1]*i)%mod,inv_jie[i]=inv(jie[i]);
    while(~scanf("%d",&n))
    {
        n--;
        for(i=0;i<=n;i++)printf("%lld ",C(n,i));
        puts("");
    }
    return 0;
}

```

## F.Zyy的生日礼物5

正确率64%

### 解法一：

因为只有五种调料，可以使用五重循环枚举每种调料的量，当他们的和为 $x$ 的时候输出答案。注意是**从大到小输出**，所以枚举的时候从6往1枚举。

```

int a, b, c, d, e, sum;
scanf("%d", &sum);
for (a = 6; a; --a)
    for (b = 6; b; --b)
        for (c = 6; c; --c)
            for (d = 6; d; --d)
                for (e = 6; e; --e)
                    if (a + b + c + d + e == sum)
                        printf("%d%d%d%d%d\n", a, b, c, d, e);

return 0;

```

### 解法二(\*有兴趣同学参考):

采用函数递归，相当于把五个循环写成同一个函数，用一个值 $x$ 来表示这是第几重循环，最后在第6循环执行前计算五个数的和判断并输出答案。

```

#include<stdio.h>
int n,a[6];
void dfs(int x)
{
    if(x==6)
    {
        int sum=0,i;
        for(i=1;i<=5;i++)sum+=a[i];
        if(sum==n)
        {
            for(i=1;i<=5;i++)printf("%d",a[i]);
            puts("");
        }
        return ;
    }
    int i;
    for(i=6;i>=1;i--)a[x]=i,dfs(x+1);
}
int main()
{
    scanf("%d",&n);
    dfs(1);
    return 0;
}

```

## G.Ix买股票

---

正确率21%

因为卖出必定在买入之后(或当天)，我们用另一个数组 $Min[]$ 来记录前面的最小值，即 $Min_i$ 表示 $a_1, a_2, \dots, a_i$ 中的最小值，然后用 $a_i - Min_i$ 来更新答案，即取 $a_1 - Min_1, a_2 - Min_2, \dots, a_n - Min_n$ 中的最小值。

**注意：**Min[0]要赋值为极大值

**另外：**大数组请定义在main()外面(建议大小10以上)，此时数组中每个元素有初值0

```

#include<stdio.h>
int n,a[1000010],Min[1000010];
int main()
{
    scanf("%d",&n);
    int i,ans=0;
    Min[0]=2147483647;
    for(i=1;i<=n;i++)
    {
        scanf("%d",&a[i]);
        Min[i]=a[i]<Min[i-1]?a[i]:Min[i-1];
        /*ÉÏÐÏàµ±ÓÚ£º
        Min[i]=min(Min[i],a[i]);
        */
        ans=a[i]-Min[i]>ans?a[i]-Min[i]:ans;
        /*ÉÏÒ»ÐÏàµ±ÓÚ£º
        ans=max(ans,a[i]-Min[i]);
        */
    }
    printf("%d",ans);
    return 0;
}

```

## H.lx的数字排序

---

正确率35%

使用数组 $sum[]$ 记录数的出现次数，即 $sum_i$ 表示 $i$ 这个数的出现次数，然后之后每次把出现最多的数输出然后把次数清零即可。

```

#include<stdio.h>
int n,sum[1010];
int main()
{
    scanf("%d",&n);
    int i,x;
    for(i=1;i<=n;i++)
    {
        scanf("%d",&x);
        sum[x]++;
    }
    while(n)
    {
        int ans,Max=0;
        for(i=0;i<=1000;i++)
            if(sum[i]>Max)Max=sum[i],ans=i;
        printf("%d %d\n",ans,Max);
        n-=Max,sum[ans]=0;
    }
    return 0;
}

```

## I.悠唯的签到题

---

正确率：84%

从1循环到 $n$ ,记录每一项的值，并累加。

建议：所有题浮点数请使用double，float精度太低(这并不代表建议所有地方都用long long而不用int)

```

#include<stdio.h>
int main(){
    double p=0,m=-1;
    int i,n;
    scanf("%d",&n);
    for(i=1;i<=n;i++){
        m*=(-1);
        p+=1.0/(2*i-1)*m;
    }
    printf("%.6f\n",p);
    return 0;
}

```

(代码来源：牟钰)

## J.min\_26

---

正确率44%

最优的方案当然是把小的数放在前面。注意**0不能放在最前面**，所以拿一个非零的最小的数放在0前面即可。

```
#include<stdio.h>
int a1[12];
int main(){
    for(int i=0;i<=9;i++)scanf("%d",&a1[i]);
    for(int i=1;i<=9;i++)
        if(a1[i]!=0){
            printf("%d",i);a1[i]--;break;
        }
    for(int i=0;i<=9;i++)
        while(a1[i]!=0){
            printf("%d",i);a1[i]--;
        }
    return 0;
}
```

(代码来源：杨周启珂)

## K.酸奶的矩阵乘法

---

正确率75%

三重循环即可，提示：新矩阵的第 $i$ 行第 $j$ 列的元素为第一个矩阵的第 $i$ 行向量与第二个矩阵的第 $j$ 列向量点乘的值

```

#include <stdio.h>
#include <stdlib.h>

int a[200][200],n,b[200][200];

int main()
{
    scanf("%d",&n);
    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
            scanf("%d",&a[i][j]);
    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
            for(int k=1;k<=n;k++)
                b[i][j]+=a[i][k]*a[n-j+1][k];
    for(int i=1;i<=n;i++)
    {
        for(int j=1;j<=n;j++)
            printf("%d ",b[i][j]);
        printf("\n");
    }

    return 0;
}

```

## L.HugeGun学姐疯了

---

正确率67%

两重循环输出，先计算出最后一行'\$'的数量 $n^2$ ，据此来确定第 $i$ 行该输出的空格数： $\frac{n^2}{2} - \frac{i^2}{2}$ 。然后根据 $i$ 的奇偶性输出'\$'和'@'即可。

```
#include<stdio.h>
int main()
{
    int n;scanf("%d",&n);
    int Max=n*n/2,i,j;
    for(i=1;i<=n;i++)
    {
        int x=Max-i*i/2;
        for(j=1;j<=x;j++)printf(" ");
        for(j=1;j<=i*i/2;j++)printf("$");
        if(i&1)printf("$");
        else printf("@");
        for(j=1;j<=i*i/2;j++)printf("$");
        puts("");
    }
    return 0;
}
```