

最后一次上机题解

----by 芜湖大司马

A.你家那块大破布

正确率只有 $\frac{1}{3}$ ，多捞啊，注意看题，0和0在本题不是相反数。

```
#include<stdio.h>
int a,b;
int main()
{
    scanf("%d%d",&a,&b);
    if(a==b&&a)printf("ni jia na kuai da po bu");
    else printf("ni jia la gua da pu bu");
    return 0;
}
```

B.酸奶打游戏1

模拟题，按要求一步一步打，注意1、治疗后血量不能超过上限(即初始血量) 2、治疗量等于对方攻击力时要投降

```
#include<stdio.h>
#include<string.h>
typedef long long LL;
int min(int a,int b){return a<b?a:b;}
int a[4],b[3];
int main()
{
    int i;
    for(i=1;i<=3;i++)scanf("%d",&a[i]);
    for(i=1;i<=2;i++)scanf("%d",&b[i]);
    a[0]=a[1];
    int ans=0;
    while(b[1]>0)
    {
        ans++;
        a[1]-=b[2];
        if(a[1]<=0||b[1]<=0)
        {
            printf("I died in round ");
            break;
        }
        ans++;

        if(a[1]<=b[2])
        {
            int jia=min(a[3],a[0]-a[1]);
            if(jia<=b[2])
            {
                printf("I print GG in round ");
                break;
            }
            else a[1]+=jia;
        }
        else b[1]-=a[2];
    }
    if(b[1]<=0)printf("Enemy died in round ");
    printf("%d",ans);
    return 0;
}
```

C.Max与神秘博士

根据提示，可以将数组倍长，以达到拼接成一个环的目的，方便枚举。

```
#include<stdio.h>
typedef long long LL;
int n,a[2020];
int main()
{
    int k;
    scanf("%d%d",&n,&k);
    int ans=0;
    int i;
    for(i=1;i<=n;i++)scanf("%d",&a[i]),a[i+n]=a[i];
    for(i=1;i<=n;i++)
    {
        int j;
        int temp=0;
        for(j=i;j<=k+i-1;j++)temp+=a[j];
        ans=temp>ans?temp:ans;
    }
    printf("%.2f",1.*ans/k);
    return 0;
}
```

D.简化比例

注意到*l*的范围很小，两重循环*i, j*枚举答案，再判断 $\gcd(i, j)$ 是否为1，再判是否满足条件。建议：1、使用乘法代替除法比较两个分数大小 2、与当前答案比较大小而不是直接按题意取 $\frac{A'}{B'} - \frac{a}{b}$ 最小

```
#include<stdio.h>
typedef long long LL;
int gcd(int a,int b)
{
    int c;
    while(a%b)
    {
        c=a%b,a=b,b=c;
    }
    return b;
}
int a,b,aa,bb,l,ans1,ans2;
int pd(int x,int y)
{
    return (!ans1&&!ans2)?1:ans1*y>=ans2*x;
}
int main()
{
    scanf("%d%d%d",&a,&b,&l);
    int i,j;
    for(i=1;i<=l;i++)for(j=1;j<=l;j++)if(gcd(i,j)==1)
    {
        if(b*i>=j*a&&pd(i,j))ans1=i,ans2=j;
    }
    printf("%d %d",ans1,ans2);
    return 0;
}
```

E.求和2

正确率17%，多捞哟。求出前缀和数组 b ， $b_r - b_{l-1}$ 即可。

```
#include<stdio.h>
typedef long long LL;
LL n,a[1000010],q,l,r,i;
int main()
{
    scanf("%lld%lld",&n,&q);
    for(i=1;i<=n;i++)scanf("%lld",&a[i]),a[i]+=a[i-1];
    for(i=1;i<=q;i++)
    {
        scanf("%lld%lld",&l,&r);
        printf("%lld\n",a[r]-a[l-1]);
    }
    return 0;
}
```

F.摸鱼助教Mogg IV

把个位，十位，百位，千位数字0到9全部打表出来，直接输出四个字符串即可。

```
#include<stdio.h>
typedef long long LL;
int n;
char s[5][11][11]={
    {},
    {"", "I", "II", "III", "IV", "V", "VI", "VII", "VIII", "IX"},
    {"", "X", "XX", "XXX", "XL", "L", "LX", "LXX", "LXXX", "XC"},
    {"", "C", "CC", "CCC", "CD", "D", "DC", "DCC", "DCCC", "CM"},
    {"", "M", "MM", "MMM"}
};
int main()
{
    while(~scanf("%d",&n))
    {
        int a=n/1000,b=n%1000/100,c=n%100/10,d=n%10;
        printf("%s%s%s%s",s[4][a],s[3][b],s[2][c],s[1][d]);
        puts("");
    }
    return 0;
}
```

G.Ity过生日

高精度减法，注意考虑 $A = B$ 的情况。尽量不要使用gets或fgets，实在要用，请使用类似下方代码中的努棒写法

```
#include<stdio.h>
#include<string.h>
typedef long long LL;
int a[111],b[111],len1,len2;
char s[111],t[111];
int main()
{
    //    fgets(s+1,105,stdin),fgets(t+1,105,stdin);
    gets(s+1),gets(t+1);
    while(s[len1+1]>='0'&&s[len1+1]<='9')len1++;
    while(t[len2+1]>='0'&&t[len2+1]<='9')len2++;
    int i;
    for(i=1;i<=len1;i++)a[len1-i+1]=s[i]-'0';
    for(i=1;i<=len2;i++)b[len2-i+1]=t[i]-'0';
    int ex=0;
    for(i=1;i<=len1;i++)
    {
        a[i]-=b[i]+ex;
        if(a[i]<0)ex=1,a[i]+=10;
        else ex=0;
    }
    while(a[len1]==0&&len1)len1--;
    if(!len1)len1++;
    for(i=len1;i>=1;i--)printf("%d",a[i]);
    return 0;
}
```

H.悠唯call悠唯2

DP。对于 n 个圆盘，枚举一个 $x(0 \leq x < n)$ ，先把 x 个移到第二或第三个柱子上，再移动后面 $n - x$ 个到第四根柱子上，最后再将前 x 个移到第四根柱子上 总共的步数，对所有 x 的答案取最小值即可。那么就转化成要算圆盘数为 x 的子问题，就可以递归计算了。当然递归只能处理 n 在20左右的数据。下面给出dp代码(递推)、递归代码、递归变为DP代码(记忆化搜索)。

dp:

```
#include<stdio.h>
typedef long long LL;
LL min(LL a,LL b){return a<b?a:b;}
int n,t;
LL a[66];
int main()
{
    scanf("%d",&t);
    while(t-->0)
    {
        scanf("%d",&n);
        a[1]=1,a[2]=3;
        int i;
        for(i=3;i<=n;i++)
        {
            a[i]=1000000000000000000LL;
            int j;
            for(j=0;j<i;j++)a[i]=min(a[i],(1LL<<j)-1+2LL*a[i-j]);
        }
        printf("%lld\n",a[n]);
    }
    return 0;
}
```


递归(TLE):

```
#include<stdio.h>
typedef long long LL;
LL min(LL a,LL b){return a<b?a:b;}
LL js(int n)
{
    if(n==0)return 0;
    LL ret=1000000000000000000LL;
    int i;
    for(i=0;i<n;i++)ret=min(ret,js(i)*2+(1LL<<(n-i))-1);
    return ret;
}
int n,t;
int main()
{
    scanf("%d",&t);
    while(t-->0)
    {
        scanf("%d",&n);
        printf("%lld\n",js(n));
    }
    return 0;
}
```

记忆化搜索:

```
#include<stdio.h>
typedef long long LL;
LL min(LL a,LL b){return a<b?a:b;}
LL a[66];
LL js(int n)
{
    if(a[n])return a[n];
    if(n==0)return 0;
    LL ret=1000000000000000000LL;
    int i;
    for(i=0;i<n;i++)ret=min(ret,js(i)*2+(1LL<<(n-i))-1);
    return a[n]=ret;
}
int n,t;
int main()
{
    scanf("%d",&t);
    while(t-->0)
    {
        scanf("%d",&n);
        printf("%lld\n",js(n));
    }
    return 0;
}
```

希望大家仔细比对后两份代码，学会将暴力递归转化为记忆化搜索(DP的一种)再转化为DP递推，当然，有时候记忆化搜索更好写

I.RyogiT捉login:

经典题目，最长上升(下降)子序列。

$n \leq 20$ ，可以直接暴力枚举每个数选不选，再判断是否严格下降，再更新答案。下面给出 2^n , n^2 , $n \log_2 n$ 的三种做法。

2^n :

```
#include<stdio.h>
#include<string.h>
typedef long long LL;
int max(int a,int b){return a>b?a:b;}
int n,a[22],b[22],Max,temp;
void dfs(int x)
{
    if(x==n+1)
    {
        int i;
        for(i=1;i<temp;i++)if(b[i]<=b[i+1])return ;
        Max=max(Max,temp);
        return ;
    }
    dfs(x+1);
    b[++temp]=a[x];
    dfs(x+1);
    temp--;
}
int main()
{
    scanf("%d",&n);
    int i;
    for(i=1;i<=n;i++)scanf("%d",&a[i]);
    dfs(1);
    printf("%d",n-Max);
    return 0;
}
```

n^2 :

```
#include<stdio.h>
#include<string.h>
typedef long long LL;
int n,a[22],sta[22],top;
int main()
{
    scanf("%d",&n);
    int i;
    for(i=1;i<=n;i++)scanf("%d",&a[i]);
    for(i=1;i<=n;i++)
    {
        if(!top||a[i]<sta[top])sta[++top]=a[i];
        else
        {
            int j;
            for(j=1;j<=top;j++)if(sta[j]<=a[i])
            {
                sta[j]=a[i];
                break;
            }
        }
    }
    printf("%d",n-top);
    return 0;
}
```

$n\log_2 n$:

```
#include<stdio.h>
#include<string.h>
typedef long long LL;
int n,a[22],sta[22],top;
int main()
{
    scanf("%d",&n);
    int i;
    for(i=1;i<=n;i++)scanf("%d",&a[i]);
    for(i=1;i<=n;i++)
    {
        if(!top||a[i]<sta[top])sta[++top]=a[i];
        else
        {
            int l=1,r=top,mid;
            while(l<r)
            {
                mid=(l+r)>>1;
                if(sta[mid]>a[i])l=mid+1;
                else r=mid;
            }
            sta[l]=a[i];
        }
    }
    printf("%d",n-top);
    return 0;
}
```

这学期的程设课程快结束了，客观来讲，本学期的程设课题目比去年难度差不多，但是题量至少是去年的两倍以上。上机题没补完？练习赛题目没补完？没关系，做不完不是你的错。但是，希望同学们在假期和平时能够加强对C/C++编程的训练，不仅仅只限于buaacoding.cn，也可以去洛谷，codevs，UVA等各大OJ做题。上学期，17级的数据结构课将很多人考自闭了，我们不希望同学们走上我们的道路，期末考试50分满分，均分16(据说)，其中填空题10分，平均无法完成一道编程题。*

*好像说得有点多，千言万语汇成6个字符："加油！"