循环 题解

A.The King of Subtraction

for循环枚举次数，按要求模拟操作

```
#include<stdio.h>
int main(){
        int n,k;
        scanf("%d%d",&n,&k);
        for(int i=1;i<=k;i++){
                if(n%10==0){
                        n/=10;
                }
                else n--;
        }
        printf("%d\n",n);
        return 0;
}
```

B.The King of Guessing

判断$2*a$和b的大小，如果前者大，则优先选择后者精力花费少，若有余数再选择a；反之选择前者。

```
#include<stdio.h>
int q,a,b;
long long n;
int main(){
        scanf("%d",&q);
        while(q--){
                scanf("%lld%d%d",&n,&a,&b);
                if(2*a>b){
                        if(n%2==1)
                             printf("%lld\n",(n/2)*b+(n%2)*a);
                        else printf("%lld\n",n/2*b);
                }
                else printf("%lld\n",n*a);
        }
        return 0;
}
```

C.The King of GCD

- 更相减损术
- 如果$A > B$，则 $gcd(A, B) = gcd(A - B, B)$
  如果$A < B$，则 $gcd(A, B) = gcd(B - A, A)$
- 简而言之，两数比较，gcd(差值，较小数)

```
#include<stdio.h>
int main(){
        int a,b;
        scanf("%d%d",&a,&b);
        while(a!=b){
                if(a>b){
                        a=a-b;
                }
                else{
                        b=b-a;
                }
        }
        printf("%d\n",a);
        return 0;
}
```

- 辗转相除法(欧几里得算法)
- 定理：两个正整数a和$b(a>b)$，它们的最大公约数等于a除以b的余数c和b之间的最大公约数。
- $gcd(A,B)=gcd(B,AmodB)$ 其中:$A>B$

```
#include<stdio.h>
int main(){
        int a,b,tmp;
        scanf("%d%d",&a,&b);
        if(a<b){  //若a<b，交换ab大小
                tmp=a;
                a=b;
                b=tmp;
        }
        while(b!=0){
                tmp=a%b;
                a=b;
                b=tmp;
        }
        printf("%d\n",a);
        return 0;
}
```

D.The King of Sequence

循环枚举n，当满足$S_n > K$的时候跳出循环

注意一个问题：用double的s和int的k直接比较，会有精度问题，double之间的比较也有坑(定义eps)

```
#include<stdio.h>
const double eps=1e-8;
int main()
{
        int i;//变量定义
        double k;
        scanf("%lf",&k);
        double s = 0;
        for (i = 1; ; i++)
        {
                s += 1.0 / i;//数列叠加，注意除法两边数据类型
                if (s - k > eps)break;//当不满足条件，跳出循环
        }
        printf("%d\n",i);//输出结果
        return 0;
}
```

若将条件写在for循环中

```c
#include<stdio.h>
int main()
{
        int i;//变量定义
        double k;
        scanf("%lf",&k);
        double s = 0;
        for (i = 1; s<=k; i++)
        {
                s += 1.0 / i;//数列叠加，注意除法两边数据类型
                //printf("s=%lf\n",s);
        }
        printf("%d\n",i-1);//输出结果
        return 0;
}
```

E.The King of Codeforces

循环枚举天数，特判周六周日

```
#include<stdio.h>
int main(){
        int x,n,ans=0;
        scanf("%d%d",&x,&n);
        for(int i=0;i<n;i++){
                if(x==6||x==7){
                        if(x==7)x=1;
                        else x++;
                        continue;
                }
                ans+=5;
                x++;
        }
        printf("%d\n",ans);
        return 0;
}
```

F.The King of Factorials

- 嵌套循环，变量用long long存储
- 每步取模，要计算只包含加法、减法和乘法的整数表达式除以正整数n的余数，可以在每步计算之后对n取余，结果不变。

```
#include<stdio.h>
const int mod=1e6;
int main(){
        int n;
        long long s=0;
        scanf("%lld",&n);
        for(int i=1;i<=n;i++){
                long long tmp=1;
                for(int j=i;j>=1;j--){
                        tmp=tmp*j%mod;
                }
                s=(s+tmp)%mod;
        }
        printf("%lld\n",s%mod);
        return 0;
}
```

若输入的n达到了10^6，程序还能在$1s$中内跑出答案吗？

改进方法一

```
#include<stdio.h>
const int mod=1e6;
int main(){
        int n;
        long long s=0,tmp=1;
        scanf("%lld",&n);
        for(int i=1;i<=n;i++){
                s=(s+tmp*i)%mod;
                tmp=tmp*i%mod;
        }
        printf("%lld\n",s%mod);
        return 0;
}
```

改进方法二

我们可知25!的末尾有6个0，30!的末尾有7个0。所以在25之后$mod10^6$都为0。

```
#include<stdio.h>
const int mod=1e6;
int main(){
        int n;
        long long s=0;
        scanf("%lld",&n);
        if(n>25)n=25;
        for(int i=1;i<=n;i++){
                long long tmp=1;
                for(int j=i;j>=1;j--){
                        tmp=tmp*j%mod;
                }
                s=(s+tmp)%mod;
        }
        printf("%lld\n",s%mod);
        return 0;
}
```

G.The King of Games

首先判断n是否能到达m，即m是否能整除n，不能整除则输出-1。否则，分别将2和3除尽并计算次数。注意特例$n=1$，$m=5$

```c
#include<stdio.h>
int main(){
	int n,m,ans=0;
	scanf("%d%d",&n,&m);
	if(m%n!=0)puts("-1");
	else{
		int tmp=m/n;
		while(tmp%3==0){
			tmp/=3;
			ans++;
		}
		while(tmp%2==0){
			tmp/=2;
			ans++;
		}
		if(tmp!=1)ans=-1;
		printf("%d\n",ans);
	}
	return 0;
}
```

H.The King of Factorials Again

```
1. 1!!=1
2. 2!!=2
3. 3!!=720
4. 4!!=620448401733239439360000
5. 5!!很大
```

所以前面的某些数($n = 0, 1, 2, 3$)需要特判，后面的数的阶乘一定能够整除mod。注意0的阶乘是1和mod为1、2的情况。

```c
#include<stdio.h>
long long n, mod;
int main()
{
	while (~scanf("%lld%lld", &n, &mod))
	{
		if ((n == 0 || n == 1) && mod == 1)
			printf("0\n");
		else if ((n == 0 || n == 1) && mod > 1)
			printf("1\n");
		else if (n == 2 && mod <= 2)printf("0\n");
		else if (n == 2 && mod > 2)printf("2\n");
		else if (n == 3)
		{
			long long tmp = 1;
			for (int i = 2; i <= 720; i++)
				tmp = (tmp * i) % mod;
			printf("%lld\n", tmp);
		}
		else printf("0\n");
	}
	return 0;
}
```

I.The King of Brute Force

思路：其他美元面值都可以用1面值美元来表示，其他欧元面值都可以用5面值欧元表示，暴力枚举欧元面值，剩下的用来交换美元，查找最少的剩余卢布数。(同理枚举美元面值，剩下的用来交换欧元，可行)

```
#include<stdio.h>
#include<iostream>
using namespace std;
int n,d,e;
int main(){
        scanf("%d%d%d",&n,&d,&e);
        int left=0,ans=n;
        for(int i=0;i*e*5<=n;i++){
                left=n-i*e*5;
                ans=min(ans,left%d);
        }
        printf("%d\n",ans);
        return 0;
}
```