

# 1273: 【基础】均匀划分

## 题目描述

有两个正整数K与P，然后将K分成P个部分： $K_1, K_2, \dots, K_p$ ，要求找出一种分法，满足： $K_1 + K_2 + \dots + K_p = K$ ，并使 $K_1 \times K_2 \times \dots \times K_p$ 为最大。例如： $K=4, P=2$ ，则分法有： $4+0=4, 3+1=4, 2+2=4$  其中乘积最大者为： $2 \times 2 = 4$

## 输入

输入两个正整数K与P ( $2 \leq K \leq 35, P < K$ )，输入不必查错。

## 输出

满足条件的最大乘积的一种分法。

## 样例输入

```
1  样例1 :
2  4 2 ( 也就是K=4, P=2 )
3
4  样例2 :
5  17 3
```

## 样例输出

```
1  样例1 :
2  2*2=4
3
4  样例2 :
5  5*6*6=180
```

## 来源/分类

[2003年江苏省小学生信息学（计算机）奥赛试题](#)

## 思路

奥数题：

当我们不知道数学原理的时候，可以用李明烨的那种搜索构建的方式，哪怕得部分分数也是好的。总之一句话，不轻言放弃！哪怕代码很长，很难写。尝试有可能成功，但是不写放弃就是0。

---

```
1  //钱嘉欢
2  #include<iostream>
3  using namespace std;
4  long long n,m,a[40],p,h;
5  int main()
6  {
```

```

7     cin>>n>>m;
8     //输入
9     for(int i=1;i<=m;i++)
10        a[i]=n/m;
11    //在a里存把n平均分成m份的值
12    p=n%m;
13    //p等于把n平均分后的余数
14    for(int i=m-p+1;i<=m;i++)
15        a[i]++;
16    //给后p个a加1,因为因数和同样,积要大,因数就要尽量相同
17    h=1;
18    for(int i=1;i<m;i++)
19    {
20        cout<<a[i]<<"*";
21        h=h*a[i];
22    }
23    h=h*a[m];
24    //h等于所有因数的乘积
25    cout<<a[m]<<"="<<h;
26    //输出
27    return 0;
28 }
29

```

```

1 //金子杨
2 #include<iostream>
3 #include<string>
4 using namespace std;
5 int n,k,ys,a[40],p,ans=1;
6 int main()
7 {
8     cin>>n>>k; p=k+1;
9     for(int i=1;i<=k;i++)
10        a[i]=n/k;
11    ys=n%k;
12
13    while(ys>0)
14    {
15        p--;
16        a[p]+=1;
17        ys--;
18    }
19    for(int i=1;i<k;i++)
20    {
21        cout<<a[i]<<"*";
22        ans*=a[i];
23    }
24    ans*=a[k];
25    cout<<a[k]<<"="<<ans<<endl;
26    return 0;
27 }
28
29 /*思路:先平均分,然后把剩下的再分。
30
31 为什么:因为差小积大*/

```

```

1 //朱桓毅

```

```

2 //1273题 答案与分析
3 #include<iostream>
4 using namespace std;
5 int p,k,x,ans=1;
6 int a[35];
7 int main()
8 {
9     /* 在和相同的情况下,
10     两个数(或更多)的差越小,积越大。*/
11
12     //如果有余数,
13     //那么就不能让每一个数相同。
14     cin>>k>>p;
15     x=k%p;
16     //在不考虑余数的情况下,
17     //让每一个数分得k
18     k=k-x;
19     k=k/p;
20     for(int i=0;i<p;i++)
21     {
22         a[i]=k;
23     }
24     //考虑余数
25     for(int i=0;i<x;i++)
26     {
27         a[i]++;
28     }
29     //计算答案
30     for(int i=0;i<p;i++)
31     {
32         ans*=a[i];
33     }
34     //输出
35     cout<<a[p-1];
36     for(int i=p-2;i>=0;i--)
37     {
38         cout<<"*"<<a[i];
39     }
40     cout<<"="<<ans;
41     return 0;
42 }
43

```

```

1 //周子涵
2 #include<bits/stdc++.h>
3 using namespace std;
4 int k,p,s,ys,ans=1;//k=数,p=组数,s=商,ys=余数,ans=答案
5 int main(){
6     cin>>k>>p;//输入
7     s=k/p;//把商求出来
8     ys=k%p;//把余数求出来
9     for(int i=1;i<=(p-ys);i++){//循环到组数-余数
10         if(i==p) cout<<s<<"=";//如果是最后一个,输出等号
11         else cout<<s<<"*";//如果不是,输出乘号
12         ans*=s;//答案记录
13     }
14     for(int i=1;i<=ys;i++){//循环到余数
15         if(i==ys) cout<<s+1<<"=";//如果是最后一个,输出等号

```

```

16         else cout<<s+1<<"*"; //如果不是，输出乘号
17         ans*=(s+1); //答案记录
18     }
19     cout<<ans; //输出
20     return 0; //因为两数越接近，乘积越大，所以我的思路是数尽可能平均。
21 }
22

```

```

1 //殷学楷
2 #include<bits/stdc++.h>
3 using namespace std;
4 long long ans; //ans保存乘积;
5 int a[100],k,p,p2; //a数组来保存每个数;
6 int main() {
7     cin>>k>>p;
8     p2=p; //防止p被修改;
9     for(int i=1;i<=p;i++){ //每一个数枚举;
10         //平均分
11         if(k%p2!=0){ //如果有余数的话，这个数加一;
12             a[i]=k/p2+1;
13             k-=a[i];
14             p2--;
15         }
16         else{ //否则的话，这个数不用加一;
17             a[i]=k/p2;
18             k-=a[i];
19             p2--;
20         }
21     }
22     ans=1; //初始化ans
23     for(int i=1;i<=p;i++) ans*=a[i]; //把每个数都乘起来;
24     cout<<a[p]; //输出第一个，保持格式;
25     for(int i=p-1;i>=1;i--) cout<<"*<a[i];
26     cout<<"<<ans; //输出答案;
27     return 0;
28 }
29

```

```

1 //李明烨 2019/5/28 20:25:39
2 #include<cstdio>
3 #include<cstring>
4 #include<iostream>
5 #include<algorithm>
6 using namespace std;
7 long long k,p,big/*最大*/,chai[1005]/*类似全排列的数组*/,ans[1005]/*存答案*/;
8 void check(int m){
9     big=big>m?big:m; //比较两者大小
10     if(big==m/*如果big被改动了*/){
11         for(int i=1;i<=p;i++){
12             ans[i]=chai[i];
13         } //把拆分方法复制一下
14     }
15     return ;
16 }
17 void dfs(int shu,int now,int mul/*long long可能会溢出*/){
18     if(now==p+1){ //如果拆完了

```

```

19         if(shu==0){//如果拆到底了
20             check(mul); //做一个比较
21             return ;
22         }
23     }
24     if(shu==0){//如果已经没了，直接跳出
25         return ;
26     }
27     for(int i=2;i<=shu;i++){//1对乘积没有帮助，从2开始，节省时间
28         chai[now]=i; //记录i
29         dfs(shu-i /*数拆掉i后*/, now+1 /*去下一个*/, mul*i /*乘积乘上*/); //转移公式
30         chai[now]=0; //清零备用
31     }
32     return ;
33 }
34 int main(){
35     scanf("%lld%lld",&k,&p); //输入
36     dfs(k /*当前拆分的数*/, 1 /*当前所在位置*/, 1 /*积, 不能是0*/);
37     sort(ans+1, ans+p+1); //从小到大处理
38     printf("%lld", ans[1]); //特殊输出
39     for(int i=2;i<=p;i++){
40         printf("%lld", ans[i]); //输出数字
41     }
42     printf("%lld\n", big); //输出答案
43     return 0;
44 }
45 /*李明烨 2019/5/28 20:29:19
46 思路：主要是建模——全排列：暴力一个一个枚举这个数可能是几，完成后进行比较乘积大小，然后输出
47 输出前还要从小到大来，不然估计会错*/
48

```