一个人在全部d天中每天坐哪辆车，可以表示为d位k进制数x。那么2个人每天都在同一辆车上，就是两个人的x相等。所以我们只要构造出n个不同的d位k进制数就行。

先判若d^k<n则无解。（注意用(int)pow(5,4)会得比实际结果小1的整数……pow还是用double来撸比较好）

然后就撸出n个d位k进制数，我是直接0~n-1，转换成k进制数。

最后输出，我是由0~k-1组成的k进制数，输出时加1就行。

注意k非常大。。pow会溢出 得判断pow的溢出 //或者自己模拟乘法,并在过程中判断溢出

可以开double解决

每一列是一个d位k进制

#include <cstdio>

#include <cstring>

#include<iostream>

#include<algorithm>

#include<bits/stdc++.h>

using namespace std;

int a[1010][1010];

int main()

{

//freopen("input.txt","r",stdin);

int n,k,d;

cin>>n>>k>>d;

if((double)n>pow((double)k,(double)d))

puts("-1");

else

{

for(int i=0;i<n;i++)

{

int x=i,j=0;

while(x)

{

a[j++][i]=x%k;

x/=k;

}

}

for(int i=0;i<d;i++)

for(int j=0;j<n;j++)

cout<<a[i][j]+1<<(j==n-1?'\n':' ');

}

return 0;

}