唯一分解定理：一个非质数一定可以分解为若干质数的乘积（质数的话则是它本身），而且这种分解是最彻底的，如果你分解的因子中还有合数，那么这个合数可以继续分解，直到所有因数都是质数。

那么，这个题目就可以转化为求质因数的问题：

# include<iostream>

# include<cstdio>

# include<cstring>

# include<algorithm>

#include<vector>

#include<cmath>

using namespace std;

int main()

{

int n,k;

int a[100010];

while(~scanf("%d%d",&n,&k))

{

int t(0);

memset(a,0,sizeof(a));

for(int i=2;i<=n;i++)

{

while(n%i==0)

{

a[++t]=i;

n/=i;

}

}

if(t<k)

printf("-1\n");

else

{

for(int i=1;i<k;i++)

printf("%d ",a[i]);

long long ans(1);

for(int i=k;i<=t;i++)

ans\*=a[i];

printf("%lld\n",ans);

}

}

return 0;

}

注意这么求直接求的就是质因数了，因为从小到大遇到每个能除尽的数，都会先一直除它，这样后面再有n的因子，那么这个因子就一定不会能除尽它的了