# Yuangeng

4992

#include <iostream>

#define LL long long int

#define ls (x << 1)

#define rs (x << 1 | 1)

#define MID int mid=(l+r)>>1

using namespace std;

const int N = 1000000+10;

int P[N],vis[N],phi[N],tot,n;

inline int gcd(int a,int b){return b==0 ?a:gcd(b,a%b);}

inline void prepare(){

phi[1]=1;

for(int i=2;i<N;++i){

if(!vis[i])P[++tot]=i,phi[i]=i-1;

for(int j=1;j<=tot;++j){

if(i\*P[j]>=N)break;

vis[i\*P[j]]=1;

if(i%P[j])phi[i\*P[j]]=phi[i]\*phi[P[j]];

else{phi[i\*P[j]]=phi[i]\*P[j];break;}

}

}

}

inline int QPow(int d,int z,int Mod){

int ans=1;

for(;z;z>>=1,d=1ll\*d\*d%Mod)if(z&1)ans=1ll\*ans\*d%Mod;

return ans;

}

inline bool check(int x)

{

if(x==2 || x==4)return 1;

if((x&1)^1)x>>=1;

for(int i=2;P[i]<=x;++i)

if(x%P[i]==0){

while(x%P[i]==0)x/=P[i];

return x==1 ?P[i]:0;

}

return 0;

}

inline int get\_rg(int fx)

{

int pt[1010],tt=0,Txd=phi[fx];

for(int i=1;P[i]\*P[i]<=Txd;++i)

if(Txd%P[i]==0){

pt[++tt]=P[i];

while(Txd%P[i]==0)Txd/=P[i];

}

if(Txd!=1)pt[++tt]=Txd;

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

if(QPow(i,phi[fx],fx)==1){

int flag=1;

for(int j=1;j<=tt;++j)

if(QPow(i,phi[fx]/pt[j],fx)==1){

flag=0;break;

}

if(flag)return i;

}

return 0;

}

inline void work(int fx){

int tt=0,pr[N];

if(fx==2){printf("1\n");return;}

if(fx==4){printf("3\n");return;}

int T=check(fx);

if(!T){printf("-1\n");return;}

int g=get\_rg(fx);//球出来的g是最小的原根

for(int i=1,k=g;i<phi[fx];++i,k=1ll\*k\*g%fx)

if(gcd(i,phi[fx])==1)//枚举判断g^i是否是原根

pr[++tt]=k;

sort(pr+1,pr+tt+1);

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

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

printf("%d",pr[tt]);

printf("\n");

}

int main()

{

prepare();

while(scanf("%d",&n)!=EOF)work(n);

return 0;

}