

大家好，请叫我水

题目大意：

千钧一发

Description

背景

在你正确回答表达式的值以后，石板又出现了新的状况，然而这次情况十分紧急，如果无法做出正确的选择，katherine 皇后很可能再也无法和 Kanari 国王在一起了。

描述

石板上突然出现 n 个奇怪的装置，Kanari 国王仔细观察发现每个装置上面都写有两个正整数，我们记第 i 的装置上的两个正整数分别为 a_i 和 b_i 。其中 a_i 表示这个装置的特征值， b_i 表示这个装置蕴含的能量值。现在请你在这 n 个装置中选择一部分出来，使得选择的任意两个装置 i 和 j ($i \neq j$) 至少满足下面两个条件之一：

①. 不存在正整数 T 使得 $a_i^2 + a_j^2 = T^2$ 。

②. a_i 和 a_j 有大于1的公因子。

你的目标就是使得选出来装置的能量值总和最大，如果你回答正确，说不定石板会按照它所说的，将 Katharon 国最珍贵的财产交出来，否则国王，皇后以及你的小命都将难保。

Input

第一行一个正整数 N 。

第二行共包括 N 个正整数，第 i 个正整数表示 A_i 。

第三行共包括 N 个正整数，第 i 个正整数表示 B_i 。

Output

共一行，包括一个正整数，表示在合法的选择条件下，可以获得的能量值总和的最大值。

Sample Input

4 3 4 5 12 9 8 30 9

Sample Output

39

HINT

$1 \leq N \leq 1000, 1 \leq A_i, B_i \leq 10^6$

题目来源： bzoj 3158

题目分析：

显然是个二分图最大匹配，很容易看出题目两个条件直接告诉我们分奇偶就好了（这题实在是太水了。。）

代码分享：

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<algorithm>
#include<cmath>
#include<cstring>
#define ll long long
#define inf 2000000000
using namespace std;
inline int read()
{
    int x=0,f=1;char ch=getchar();
    while(ch<'0' || ch>'9'){if(ch=='-')f=-1;ch=getchar();}
    while(ch>='0'&&ch<='9'){x=x*10+ch-'0';ch=getchar();}
    return x*f;
}
int n,cnt=1,T,ans;
int a[1005],b[1005],last[1005],cur[1005];
int h[1005],q[1005];
struct edge{
    int to,next,v;
}e[100005];
int gcd(int a,int b)
{
    return b==0?a:gcd(b,a%b);
```

```

}
void insert(int u,int v,int w)
{
    e[++cnt].to=v;e[cnt].next=last[u];last[u]=cnt;e[cnt].v=w;
    e[++cnt].to=u;e[cnt].next=last[v];last[v]=cnt;e[cnt].v=0;
}
bool jud(ll x,ll y)
{
    ll t=x*x+y*y,sq=sqrt(t);
    if(sq*sq!=t)return 1;
    if(gcd(x,y)>1)return 1;
    return 0;
}
void build()
{
    for(int i=1;i<=n;i++)
        if(a[i]%2==1)insert(0,i,b[i]);
        else insert(i,T,b[i]);
    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
            if((a[i]%2==1)&&(a[j]%2==0))
                if(!jud(a[i],a[j]))insert(i,j,inf);
}
bool bfs()
{
    int head=0,tail=1;
    for(int i=0;i<=T;i++)h[i]=-1;
    q[0]=0;h[0]=0;
    while(head!=tail)
    {
        int now=q[head];head++;
        for(int i=last[now];i;i=e[i].next)
            if(e[i].v&&h[e[i].to]==-1)
            {
                h[e[i].to]=h[now]+1;
                q[tail++]=e[i].to;
            }
    }
    return h[T]!=-1;
}
int dfs(int x,int f)
{
    if(x==T)return f;
    int w,used=0;
    for(int i=cur[x];i;i=e[i].next)
        if(h[e[i].to]==h[x]+1)
        {
            w=dfs(e[i].to,min(f-used,e[i].v));
            e[i].v-=w;e[i^1].v+=w;
            if(e[i].v)cur[x]=i;
            used+=w;if(used==f)return f;
        }

    if(!used)h[x]=-1;
}

```

```
        return used;
    }
    void dinic()
    {
        while(bfs())
        {
            for(int i=0;i<=T;i++)
                cur[i]=last[i];
            ans-=dfs(0,inf);
        }
    }
    int main()
    {
        n=read();T=n+1;
        for(int i=1;i<=n;i++)a[i]=read();
        for(int i=1;i<=n;i++)b[i]=read(),ans+=b[i];
        build();
        dinic();
        printf("%d",ans);
        return 0;
    }
```