In [110]:

```
#task1

def get_nod(a,b):
    while b!=0:
        a,b=b,a%b
    return a

def startegy_from_atob(a,b,N,name1,name2):
    x=0
    y=0
    otv_log=[]
    while (x!=N and y!=N):
        if (y==b):
            y=0
            otv_log.append(name2+">")
        if (x==0):
            x=a
            otv_log.append(">"+name1)


        if (x+y>b):
            x,y=x+y-b,b
            otv_log.append(name1+">"+name2)

        else:
            x,y=0,x+y
            otv_log.append(name1+">"+name2)
    return otv_log

def solution(a,b,N):
    if (a==N):
        return ['>A']
    if (b==N):
        return ['>B']
    if (N>max(a,b) or N%get_nod(a,b)!=0):
        return list()
    res1=startegy_from_atob(a,b,N,'A','B')
    res2=startegy_from_atob(b,a,N,'B','A')

    if (len(res1)<len(res2)):
        return res1
    else:
        return res2

a=int(input())
b=int(input())
N=int(input())
otv=solution(a,b,N)



if(len(otv)==0):
    print("Impossible")
else:
    for el in otv:
        print(el,end='\n')
```

```
3
5
1
>A
A>B
>A
A>B
```

In [111]:

```
import math


my_dict=dict()

def get_max_degNa(N,a):
    if (N,a) in my_dict:
        return my_dict[(N,a)]
    if (N<a):
        return 0
    if (N%a!=0):
        return 0

    otv=1+get_max_degNa(N//a,a)
    my_dict[(N,a)]=otv
    return otv

def get_max_degN(N):
    otv=0
    for i in range(2,int(math.sqrt(N))+1):
        tek_deg=get_max_degNa(N,i)
        otv=max(otv,tek_deg)
    return otv

N=int(input())
otv=get_max_degN(N)
print(otv)
```

```
90
2
```

In [112]:

```python

A,B,C=tuple(map(int,input().split()))
N=int(input())
d=list(map(int,input().split()))

INF = 10000000
border=[INF for i in range(B+1)]
border[0]=-1
for c in range(1,B+1):
    for i in range(N):
        if (c==d[i]):
            border[c]=i
            break
        if (border[c-d[i]]<i):
            border[c]=i
            break

#print(border)
has_sum_d_greaterCAlessB=0
otv=[]
for c in range(C-A+1, B+1):
    #print(border[c])
    if border[c]!=INF:
        has_sum_d_greaterCAlessB=1
        tek_c=c
        while(tek_c!=0):
            #print("old",tek_c)
            otv.append(border[tek_c])
            tek_c=tek_c-d[border[tek_c]]
            #print("new",tek_c)
        break
if (has_sum_d_greaterCAlessB==0):
    print(A+B)
    print(-1)
else:
    sum_di=0
    for i in otv:
        sum_di+=d[i]
    print(A+sum_di)
    print(len(otv),end=' ')
    for el in sorted(otv):
        print(el+1,end=' ')
```

```
10 17 25
5
2 7 5 3 7
26
3 1 2 5
```

In [ ]: