```
/*
Name: Irene Huang
Team: STEMnArts
Division: Junior
Contest 4
*/

#include <bits/stdc++.h>

using namespace std;


bool sqr(int n)
{
   for(int i=3;i<=sqrt(n);i++)
   {
      if(n==i*i)
      {
         return true;
      }
   }
   return false;
}

bool pri(int n)
{
   for(int i=2;i<=sqrt(n);i++)
   {
      if(n%i==0)
      {
         return false;
      }
   }
   return true;
}
int main()
{
   int board[53]={0},p1,p2,p3,me,times,roll,row;

   cin>>p1>>p2>>p3>>me>>times;

   board[p1]=-1;
   board[p2]=-1;
   board[p3]=-1;
   for(int i=0;i<times;i++)
   {
      bool found=false;
      cin>>roll;
```

```
        if(me+roll==52)
        {
           cout<<"GAME OVER";
           break;
        }
        else if(me+roll<52 && board[me+roll]!=-1)
           me+=roll;

        if(sqr(me))
        {
           for(int j=1;j<=6;j++)
           {
           if(board[me-1]!=-1)
           {
            me=me-1;
           }
           }
        }
        else if(pri(me))
        {
        for(int j=1;j<=6;j++)
           {
           if(board[me+1]!=-1)
           {
           me=me+1;
           }
           }
        }
        else if(me-roll<=26 && me>=28){
            for(int i=me-roll+1;i<=me;i++)
              {
                 if(i%roll==0 && board[i]!=-1)
                 {
                    me=i;
                    found=true;
                    break;
                 }
              }
            if(!found)
            me=me-roll;
           }
        else if(me-roll<=22){
           row=((me-3)/5)-((me-roll-3)/5);
           if(((me-roll)%5!=2 && row>0) || (me-roll)%5==2 && roll==6)//not at the corner and move
rows
           {
           for(int i=me-roll+1;i<=me;i++)
              {
                 if(i%roll==0 && board[i]!=-1)
```

```
                {
                    me=i;
                    found=true;
                    break;
                }
            }
           if(!found)
           me=me-roll;
          }
       }
       else if(me-roll>=31)
       {
       row=((me-1)/5)-((me-roll-1)/5);
          if(((me-roll)%5!=0 && row>0) || (me-roll)%5==0 && roll==6)//not at the corner and move
rows
          {
          for(int i=me-roll+1;i<=me;i++)
             {
                if(i%roll==0 && board[i]!=-1)
                {
                    me=i;
                    found=true;
                    break;
                }
             }
           if(!found)
           me=me-roll;
          }
       }

   }
 cout<<me<<endl;
   return 0;
}
```