```
/*
Name: JAY PANG

Team: STEMnArts

Division: Junior

Contest 4
*/


#include <iostream>
#include <iomanip>

using namespace std;

bool IsPrime(int);
bool IsSqure(int);
bool Rule9(int, int);
int main()
{

int p = 5;

int en1[5], en2[5], en3[5], pos[5];
int r[5];
bool moving = false;
bool prime = false;
bool square = false;
bool special = false;
int s=0;
int* dice[5];
// input
for(int t = 0; t < 5; t++)
{
   cin>>en1[t]>>en2[t]>>en3[t]>> pos[t]>> r[t];
   dice[t] = new int[r[t]];
   for(int i=0; i<r[t]; i++)
   {
      cin>>dice[t][i];
   }
}
//loop
for(int t = 0; t<5; t++)
{

   s = 0;
```

```
for(int i = 0; i<r[t]; i++)
{
   s = pos[t] + dice[t][i];
   //rule 6
   if(s == 52)
   {
      pos[t]=-1;
      break;
   }

   if (s>52)
   {
      pos[t]= pos[t]+0;
   }
   else
      //rule 4
   {
      if( s != en1[t] && s!= en2[t] && s!= en3[t])
      {
//rule 7 prime
         moving = true;
         if ( IsPrime(s) )
            {
               prime = true;
               pos[t] = s;

               for(int j = 1; j <=6; j++)
               {
                  if(pos[t] + 1!= en1[t] && pos[t] + 1!= en2[t] && pos[t] + 1!= en3[t])
                     {
                        pos[t] = pos[t] +1;

                     }
                  else
                  {

                     break;
                  }

               }
            }
//rule 8 perfect square
         else if ( IsSqure(s))
            {
                  square = true;
                  pos[t] = s;
                  for ( int j =6; j > 0; j--)
                  {
                     if(pos[t] -1 != en1[t] && pos[t] -1!= en2[t] && pos[t] -1!= en3[t])
```

```
                        {
                            pos[t] = pos[t] -1;
                        }

                    else
                    {
                        break;
                    }
                }
            }
    // rule 9 horizontal then vertical, non prime or perfect square
            else if( Rule9(s, pos[t]))

                {
                    special = true;
                    for( int j = 0; j<=dice[t][i]; j++)
                    {

                        if( (s - j)%dice[t][i]==0 && s - j!= en1[t] && s - j!= en2[t] && s - j!= en3[t] )
                        {
                            pos[t] = s - j;
                            break;
                        }
                    }
                }
// not prime, perfect square, or went horizontal and vertical
            else
            {
                pos[t] = s;
            }

        }
// lands on opponent marker
        else
        {
            pos[t] = pos[t] + 0;
        }

      }
    moving = false;
    prime = false;
    square = false;
    special = false;

    }

}

for(int t= 0; t<5; t++)
```

```
{
   if (pos[t]==-1)
   {
      cout<<t+1<<". GAME OVER"<<endl;
   }
   else{
      cout<<t+1<<". "<<pos[t]<<endl;
   }

}

return 0;
}

// rule 7 prime
bool IsPrime(int s)
{
   if ( (s% 2 != 0 && s % 3 != 0 && s % 5 != 0 && s % 7 != 0) || s == 2 || s == 3 || s == 5 || s == 7 )
   {
      return true;
   }
   else
   {
      return false;
   }
}

//rule 8 perfect square
bool IsSqure(int s)
{
   if ( s== 9 || s== 16 || s == 25 || s == 36 || s == 49)
   {
      return true;
   }
   else
   {
      return false;
   }
}

//rule 9 horizontal then vertical, not prime or perfect square
bool Rule9(int s, int pos)
{
   if((pos < 7 && s >7) || (pos < 12 && s>12) || (pos < 17 && s> 17) ||(pos < 22 && s>22) ||  (pos < 27
&& s> 27) || (pos < 35 && s>35) ||(pos < 40 && s>40) || (pos < 45 && s>45) || (pos < 50 && s>50))
   {
      return true;
   }
   else
```

```
    {
        return false;
    }
}
```