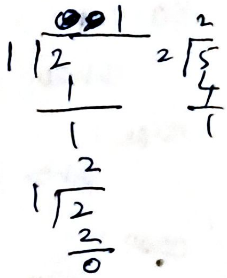
DP → Divisor Game

learning curve:

Alice will start the game first

```
if  n==0, return false
if  n==1, return false
if  n == 2, return true   (∵ Alice will choose 1).
```



$0 < x < 3$.

If Alice can't make ~~$~~ moves, she will loose the game

value = 1

~~total = n~~

~~for ((int i = value; value < n; i++){~~
~~if (n % i == 0){~~
~~value =~~

```
count = -1
for (int i = 1 ; i < n ; i++){
    if ((n % i) == 0)){
      count += 1
      n = n - i
      continue;

if (count % 2 == 0) {
      return true; }
else {
      return false;
}
```

if count is even at the end, Alice wins. If it's odd then Bob will win.

when coding = I realized i could be having repeated values and with a for-loop, we are kind of neglecting that factor. In that case, I decided to do a while loop instead to consistently ensure that the value of our n is in fact less than 0.