Homework 4

Sending lamports

Using the command line tool, send your colleagues some lamports. Check your balance and your colleagues balance before and after.

Break Solana Game

Break Solana Game
In your teams try the Break Solana game on one of the Test networks. Post a screen shot of your team's highest score on Discord.

Fizz buzz program

1. Create a project called bootcamp using Cargo
2. The main function should print a welcome message.
3. Write a 'fizz buzz' function that will be called from your main function.

1. The function should have a loop counting up to 301
2. If the count is divisible by 3, print "fizz"
3. If the count is divisible by 5 print "buzz"
4. If the count is divisible by 3 and 5 print "fizz buzz"
5. At the end print the number of times "fizz buzz" occurred.

Extra Credit

If the Fizz Buzz program was no sweat here's an additional challenge for you to have fun with.

Two Sum

We have Vector of integers called `nums` and a `target` integer.
Return the two indices that add up to the `target` value.

Rules:

- There's always one unique solution for each list.
- You can't use the same number twice.

Example 1:

```
Input: nums = [2,7,11,15], target = 9
Output: [0,1]
Explanation: Because nums[0] + nums[1] == 9, we return [0, 1].
```

Example 2:

```
Input: nums = [3,2,4], target = 6
Output:[1,2]
```

Example 3:

```
Input: nums = [3,3], target = 6
Output: [0,1]
```

```rust
fn two_sum(nums: Vec<i32>, target: i32) -> Vec<i32> {

    // your code goes here
```

```
}

fn main() {

    println!("{:?}", two_sum(vec![2, 3, 4, 5,], 9));
}
```

There's a brute force solution that's a bit easier to figure out,
but see if you can also use a HashMap for a more efficient solution.

Try it on Rust Playground