

## Things are pretty quiet at the Head First Lounge

There's a problem with the Juke Box at the Head First Lounge. The guys were working on some of the playlist code and now it's broken...

The playlist is stored as an array of strings like this:

```
playlist = ["New York, New York",
            "When You're Drinkin'",
            "Fly Me To The Moon",
            ... ← There's a whole load more
            "Mac The Knife"] tracks here.
```



The playlists can get *real long* and sometimes customers like to skip ahead by a few tracks using the `skip_tracks()` function:

```
skip_tracks(playlist, 2) ← This will skip past the first
                        2 tracks in the array.
```

When the function's called the playlist changes to something like this:

```
["Fly Me To The Moon",
 ...
 "Mac The Knife",
 "New York, New York", ← These two tracks have moved
 "When You're Drinkin'"] ← to the end of the array.
```

The function has moved the first couple of tracks to the end of the array. The function should be able to skip by any number of tracks. It should even be able to take negative numbers and skip backwards if someone wants to play some songs again.

### So what's the problem?



# Ruby Magnets

It looks like the guys had one too many tequilas when they were designing the code on the fridge door last night. A lot of the code has gotten mixed up. Do you think you can replace the magnets and rebuild the code? **You might not need to use all the magnets.**

```

def skip_tracks(arr, i)
  start = inc = i % arr.size()
  count = 1

  while inc ..... 0 && count ..... arr.size() do
    c = start

    .....

    before = .....

    ....., arr[c] = arr[c], .....

    .....

    c = .....

    .....

    .....

  end
end
end

```

Scattered code fragments (magnets):

- end
- (c + inc) % arr.size()
- arr[before]
- (c - inc) % arr.size()
- count = count + 1
- count = count + 1
- start = start - 1
- arr[before]
- <
- (c + inc) % arr.size()
- c + 1
- until
- != start
- begin
- while
- >



## Exercise

Don't take your thinking caps off too soon, because there's still more to do. Once you've rebuilt the old code, the guys are looking to make a few improvements. The old code worked, but it wasn't very... Rubyish.

So it's time to dust off the keyboard, crank your brain up to 11 and write an **entirely new version** of the code. You need to write your own version of the `skip_tracks()` function, that's shorter and neater.

Remember: the shorter the code, the better.

**It's time to get that joint jumping...**