

# Understanding

- Create a program called wordShift.cpp
  - It should take a given word and be able to do the following
    - Reverse it and print
    - Shift left n digits and print
    - Shift right n digits and print
    - quit

## Design

```
function reverse(takes a char array, or c string)
{
```

```
    take each character from the string and put it in a sequentially backwards position in a
    temporary string
```

```
    copy the temporary c string back to the referenced c string
```

```
}
```

```
function shift left (takes a c string reference and num shift)
```

```
{
```

```
    if the shift number is equal to string length
        do nothing
```

```
    if the shift number is less than string length
        copy first n characters to n offset in new array
        copy last characters from the beginning of new array
        copy from temp array to old array
```

```
    if the shift number is greater than string length
        remove string length from shift number until it is less than
        call shift left function again
```

```
}
```

```
function shift right (takes a c string reference and and num shift)
```

```
{
```

```
    if the shift number is equal to string length
        do nothing
```

```
    if the shift number is less than string length
        call shift left with inverse of number passed
```

```
    if the shift number is greater than
        reverse with method above and then call inverse shift left
```

```
{
```

```
function main
```

```
{
```

```
    get input string
```

loop on commands, parsing the first character and the remaining numbers with boilerplate code that isn't worth printing here.

}

## Testing

Input	Expected output	Actual output
perpendicular	command:	command:
L5	ndicularperpe	same
rev	eprepralucidn	same
F6	Invalid	same
R21	ralucidneprep	same
rev	perpendicular	perpendicular
R0	perpendicular	same
L0	perpendicular	same
quit	exit	exit

### 1. What did you learn?

1. Again, nothing was really new for me except some c++ related syntax. I do software development for a living, so arrays are pretty easy. I had learned about pointers on my own before this class, though this is my first time using them in projects.

I also wrote the grades program in rust, a new programming language. Check it out!

### 2. Was your understanding complete at the start of the project.

1. I believe my understanding was complete at the start of the project.

### 3. Was your initial design adequate

1. Besides minor tweaks, yes it was.

### 4. Did your tests work as expected

1. Yes. I just modified the provided test and it worked fine.

### 5. Did implementation go without any problems

1. Yes. I was happy to figure out that my left shift covered my right shift. I only had to write an algorithm once.

### 6. What techniques have we covered this week and in past weeks that helped you approach the problem, did you need to use outside sources to help solve the problem (list sites, books, or other materials that were helpful)? Does this project seem related to previous projects and do you see any names for future projects that it might be related to?

1. I was able to reuse code I wrote from last week. Otherwise, I didn't use any other sources.