8 Taxicab

8.1 Converting a String

You can use the conversion function `atoi(s)` to convert the string `s` into an int value. To use `atoi`, `<stdlib.h>` should be included. The following code shows how to use `atoi`:

```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    char *s[] = {"100", "-12"};
    int sz = sizeof s / sizeof s[0], sum = 0;

    for (int i = 0; i < sz; i++) {
        sum += atoi(s[i]);
    }
    printf("sum = %d\n", sum);

    return 0 ;
}
```

Note that there are additional conversion functions in `<stdlib.h>` such as `atol` for `long`, `atoll` for `long long`, and `atof` for `double`.

8.2 Programming Lab 8: taxicab.c

Given a sequence of instructions for the taxicab driven by an AI robot, find the final position and orientation of the taxicab. An instruction is one of the following characters: `F` (forward), `B` (backward), `L` (turn left), and `R` (turn right). The first two instructions (`F` and `B`) can have an additional integer argument n ($0 < n < 10$) denoting the number of unit distances for the movement. The turning instructions (L and R) are always turning by 90°. The initial location of the taxicab is the origin, $(0, 0)$, and it is originally directed to the east (the x-direction).

For example, given the following sequence of instructions

`F 5`, `L`, `F 2`, `L`, `B 3`, `R`, and `F 1`,

the final location is $(8, 3)$ and the orientation is the north (the y-direction).

The input consists of one or more lines in standard input. Each line contains a single instruction for the taxicab. For instructions `F` and `B`, the argument n is separated by space. Your program should print two lines in standard output. The first line should contain the x and y coordinates separated by space, and the second line, the final orientation in one of the characters in "`NEWS`": `N` for north, `E` for east, `W` for west, and `S` for south.

Additional requirements for bonus points

- Do not use `if` statement but a `switch` statement with the minimal number of `break` statements inside.
- Use `enum` constants to represent the orientation of the taxicab.

Input	**Output**
F 5 L F 2 L B 3 R F 1	8 3 N
F 1 R R F 1	0 0 W