Programming Assignment: Glowworm

Computer Science I at UCF

In this assignment, I wrote a *main()* function that could process command line arguments – Those parameters were passed to my *main()* function as an array of strings.

In this program, I print to the console an adorable little glowworm as it eats its way through an input string passed to my program. My program received two input parameters to start with: an integer indicating the maximum length of the glowworm, and a string of characters for the glowworm to eat. Those input parameters were passed as command line arguments.

The glowworm always starts with three characters: a tail (‘~’), a head (‘G’), and a big segment between the tail and the head (‘O’). Overall, a new glowworm always starts out looking like this: “~OG”

Next, process each character in the input string. The table below shows how the glowworm should react to each possible character in an input string. Following the table is a description of the output that my program should produce for each of these glowworm actions/behaviors.

Character	Description
'o' (lowercase letter o) 'O' (uppercase letter o) '@' (at symbol)	Any of these characters will cause your glowworm to grow a new segment.
's' (lowercase letter s) 'S' (uppercase letter s)	Any of these characters will cause your glowworm to shrink by one segment.
'-' (minus sign) '=' (equal symbol)	Any of these characters will cause your glowworm to inch forward.
'x' (lowercase letter x) 'X' (uppercase letter x) '%' (percent sign)	Any one of these characters will cause your glowworm to die. SAD.

Any character not listed in the table above should cause your glowworm to chill.

Side Note:

> There are certain characters that cause Linux to do wonky things with command line arguments. Accordingly, the professor guaranteed that the following characters will never appear in the command line arguments passed to the program: dollar sign (‘$’), opening parenthesis (‘(‘), closing parenthesis (‘)’), exclamation point (‘!’), hash (‘#’), ampersand (‘&’), backslash (‘\’), pipe (‘|’), semi-colon (‘;’), tilde (‘~’), asterisk (‘*’), period (‘.’), space (‘ ’), single quotes (‘ and ’), double quotes (“ and ”), and redirection symbols (‘<’ and ‘>’).

Here's how each of the glowworm behaviors listed above should work:

Growing

If the glowworm eats a character that causes it to grow, it should gain a single 'o' segment between its tail and its capital 'O' segment, causing all other segments to move forward. For example, the following output starts with an existing glowworm and shows exactly what the output should be when it grows by one segment:

```
~OG
=======

Glowworm grows:
~oOG
=======
```

A similar thing happens even if the glowworm has inched forward to another position on the platform:

```
  ~OG
=======

Glowworm grows:
  ~oOG
=======
```

If the glowworm is already maximally long, then consuming a growth character should cause it to chill instead:

```
~ooooOG
=======

Glowworm chills:
~ooooOG
=======
```

Shrinking

If the glowworm eats a character that causes it to shrink, it should lose an 'o' segment, causing all segments after that one to pull backward. For example, the following output starts with an existing glowworm and shows exactly what the output should be when it shrinks by one segment:

```
~ooOG
=======

Glowworm shrinks:
~oOG
=======
```

If the glowworm has no lowercase ‘o’ segments, then it should chill instead:

```
   ~OG
=======

Glowworm chills:
   ~OG
=======
```

Inching Forward

If the glowworm eats a character that causes it to inch forward, all segments should move forward in tandem. For example, the following output starts with an existing glowworm and shows exactly what the output should be when it inches forward:

```
~ooOG
=======

Glowworm inches forward:
 ~ooOG
=======
```

Dying

If the glowworm eats a character that causes it to die, the “OG” in the glowworm string should be replaced with “Xx” (an uppercase ‘X’ followed by a lowercase ‘x’), and the program should then terminate without processing any remaining letters in the string that the glowworm was consuming:

```
  ~ooOG
=======

Glowworm meets its demise. SAD.
  ~ooXx
=======
```

Chillin’

If the glowworm consumes a character not associated with one of the behaviors/actions above, the glowworm should not move or change in any way, and you should indicate that the glowworm is chillin’, like so:

```
 ~ooOG
=======

Glowworm chills:
 ~ooOG
=======
```

Magical, Translocational Glowworm Shenanigans

PLOT TWIST! This is some sort of magical, translocational glowworm we're dealing with. If it reaches the end of the platform it's on and then grows or inches forward, it wraps back around to the beginning of the platform. The following output examples illustrate how to handle wrap-around situations with the glowworm.

If the glowworm has grown and/or inched forward to the end of the platform and then consumes a character that causes it to grow, its head should wrap back around to the beginning of the platform:

```
  ~ooOG
=======

Glowworm grows:
G ~oooO
=======
```

From there, if the glowworm grows again, the behavior is as follows:

```
G ~oooO
=======

Glowworm grows:
OG~oooo
=======
```

Here's how shrinking affects a glowworm that has already wrapped around to the start of the platform:

```
OG  ~oo
=======

Glowworm shrinks:
G   ~oO
=======
```

Here's another example. Notice that the tail always remains stationary when the glowworm shrinks:

```
G   ~oO
=======

Glowworm shrinks:
    ~OG
=======
```

Here’s an example in which a wrapped-around glowworm inches forward:

```
OG ~ooo
=======

Glowworm inches forward:
oOG ~oo
=======
```

A wrapped-around glowworm can inch forward even if it is taking up the entire platform:

```
oOG~ooo
=======

Glowworm inches forward:
ooOG~oo
=======
```

Compilation and Running at the Command Line

```
gcc Glowworm.c
```

After compiling, the user can attempt different number and string combinations

```
./a.out 5 soso
```

```
.a.out 5 sobsob
```

```
.a.out 5 sobsxob
```

```
.a.out 7 @ooOs--Ss-------oO@-----@s-sSsS
```

```
.a.out 7 Oo--S---O===Oo------@
```