Coding Assignment 4

The format and content of the output is not a suggestion – it is the specification given to you to follow so please follow. Points will be lost for not following the specification. This includes using the specified functions. This is essential to the grading process.

Create an ASCII drawing tool.

1. Please see the examples in this document to see how tool runs and how it reacts to valid and invalid input.

2. This assignment includes using two provided files – `DrawTool.h` and `DrawTool.c`. You are required to use these files and you are not allowed to alter them. You must write your code to use these library files. You will not be submitting these two files as part of your assignment, so if you alter them to make your code work, then your code will fail when the GTA compiles with the original versions.

3. Use the following pseudocode to get you started on the program.

   Include "`DrawTool.h`" in your `Code4.c` file.

   In `main()`

   Declare a 2D array with a max size of 20 for both dimensions. Anything larger than 20 tends to wrap and be messy on the screen. This max size is set as a `#define` in `DrawTool.h`. You should not hardcode 20 anywhere in your program – use the `#define` from `DrawTool.h`.

   Call function `InitializeMap` from `DrawTool`

   Parameters

   array

   user-chosen size of the map (pass by reference).

   Call function `PrintInstructions()` from `DrawTool`

   No parameters

   Call function to `PrintMap()` from `DrawTool`

   Parameters

   array

   user-input size of the map

   Prompt for a draw command and read it using `fgets()`. Then use `strtok()` to parse out **just** the draw command (Q, P, V or H). If the user entered 'Q' or 'q' as the draw command, then your program should quit. If the user did not enter Q, then use `strtok()` to parse out the rest of the components of the draw command. If the mark value is not entered, then 'X' should be used as a default value.

   The draw command should be entered as described in the instructions. Each draw command is one prompt – not multiple prompts.

   All draw commands should be validated to ensure they will not go out of bounds. Check for out of bounds for both the coordinate and line. You cannot draw a line that will end of out of bounds.

   For the point command ('P'), update that point in the array with the input mark.

For the horizontal ('H') and vertical ('V') commands, call function `DrawLine()` from `DrawTool`. Drawing a line means marking those spots with the input mark in the array. This function should be passed the array, the row,col from the command, the action (H or V), the number of spots to mark and the mark itself. The function will use for loops to appropriately move through the array and mark the spots.

Any draw command that do not start with Q, P, V or H will generate an output of "That draw command is unknown".

Draw commands of P, V and H can be entered in upper or lowercase. Use function `toupper()` to convert all input commands to uppercase.

The program will continue to prompt for draw commands and display the array until the user quits.

4. Create the `makefile` to compile `Code4.c` and `DrawTool.c`. You are required to use the `makefile` template provided with this assignment. Be sure to include your name in the first line comment (comments in `makefile`s start with #)

5. **Files to submit in a zip file named "`Code4_xxxxxxxxxx.zip`"**

    1. `Code4_xxxxxxxxxx.c`
    2. Submit a file that you created named "`input.txt`" that contains the draw commands to output your initials. Be sure to state in your assignment submission what those initials are. Your input file needs to contain ALL of the commands to complete a full run. See video attached to assignment for how to use this file and how your program should behave. Test this process using the UNIX redirect command as shown in the video.
    3. `makefile`

ATTENTION : Do not alter the `DrawTool.c` or `DrawTool.h` files. You are not submitting those; therefore, any changes you make to them to make your code work will not be present when your code is graded.

Miscellaneous Notes

1. Array is statically allocated using a define set to the maximum value (20). The user will be prompted for what size array they want to display within that 20x20 array. When passing the array, you must use the max size, not the user input size in function calls/definitions.
2. The GTA will be running your program with your file and a file of draw commands that are both valid and invalid to test the functionality of your program. Be sure to test for and reject commands that go out of the bounds of the array. Be sure to test that you can draw along the edges of your array.
3. Using the following command

    ```
    ./Code4_xxxxxxxxxx.e < input.txt
    ```

    is using the UNIX redirect command. The < symbol takes the contents of `input.txt` and dumps it completely into `stdin`. Your program then reads from `stdin` for each prompt rather than asking you. During this process, the actual commands do not show on the screen since they are not being typed.

    It will be very important to this process that your file have UNIX end of lines rather than Mac.

    If you are on a Mac, you will need to run your input file through this UNIX command to transform the Mac CR EOLs to UNIX LF EOLs.

    ```
    cat file.txt | tr '\r' '\n' | tr -s '\n' > newfile.txt
    ```

    `file.txt` is the original file and `newfile.txt` is the new file with UNIX LF EOLs. If you do not do this step, your program will not behave correctly because it will try to process the `\r` symbol left by Mac at the end of each line.

GET YOUR PROGRAM WORKING **BEFORE** YOU TRY THE REDIRECT AND INPUT FILE. There is nothing you need to add to the program to make it work with the input file. You are not opening the file, you are not coding anything to handle a file. We are simply using the UNIX redirect command '<' to give input to the program from a file rather than the user manually typing it. Your code is the **SAME** whether you are typing in the draw commands or reading them from a file.

Sample input.txt file – you need to create your own file with your own initials

```
15
-
V(0,0,10)D
H(0,1,2)D
H(9,1,2)D
V(1,3,8)D
V(0,5,10)M
V(0,9,10)M
P(1,6,1)M
P(2,7,1)M
P(1,8,1)M
V(0,11,10)F
H(0,12,3)F
H(4,12,2)F
Q
```

Instructions

```
Draw commands start with

        P for a single point
        H for a horizontal line
        V for a vertical line

After the P/V/H, enter a row,col coordinate and the number of spots to mark
enclosed in () and separated by commas and then the character for the mark.
'X' will be used if a mark is not entered.  For example,

        P(2,3,1)*       start at point 2,3 in the array and mark one spot
                        with an *.  For P, the 3rd parameter is ignored.

        V(1,2,3)$       start at point 1,2 in the array and mark the next
                        3 spots down from the current position with $

        H(4,6,7)#       start at point 4,6 in the array and mark the next
                        7 spots to the right with #

Coordinates out of range and lines drawn past the borders are not allowed.

Enter Q at the draw command prompt to quit

Press <ENTER> to continue
```

Running program with one type of each command

```
student@maverick:/media/sf_VM/CSE1320/CA4$ ./Code4_1000074079.e
How big is the array? (Enter a value between 1 and 20) 17
What is the background character? .

Draw commands start with

        P for a single point
        H for a horizontal line
```

V for a vertical line

After the P/V/H, enter a row,col coordinate and the number of spots to mark enclosed in () and separated by commas and then the character for the mark. 'X' will be used if a mark is not entered. For example,

P(2,3,1)*	start at point 2,3 in the array and mark one spot with an *. For P, the 3rd parameter is ignored.
H(1,2,3)$	start at point 1,2 in the array and mark the next 3 spots to the right with $
V(4,6,7)#	start at point 4,6 in the array and mark the next 7 spots down from the current position with #

Coordinates out of range and lines drawn past the borders are not allowed.

Enter Q at the draw command prompt to quit

Press <ENTER> to continue

```
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .

Enter draw command (enter Q to quit) q

student@maverick:/media/sf_VM/CSE1320/CA4$ ./Code4_1000074079.e
How big is the array? (Enter a value between 1 and 20) 14
What is the background character? *

Draw commands start with

        P for a single point
        H for a horizontal line
        V for a vertical line

After the P/V/H, enter a row,col coordinate and the number of spots to mark
enclosed in () and separated by commas and then the character for the mark.
'X' will be used if a mark is not entered.  For example,

        P(2,3,1)*       start at point 2,3 in the array and mark one spot
                        with an *.  For P, the 3rd parameter is ignored.

        H(1,2,3)$       start at point 1,2 in the array and mark the next
                        3 spots to the right with $

        V(4,6,7)#       start at point 4,6 in the array and mark the next
                        7 spots down from the current position with #

Coordinates out of range and lines drawn past the borders are not allowed.

Enter Q at the draw command prompt to quit

Press <ENTER> to continue
```

```
*  *  *  *  *  *  *  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *  *  *  *  *  *

Enter draw command (enter Q to quit) V(1,1,10)H

*  *  *  *  *  *  *  *  *  *  *  *  *  *
*  H  *  *  *  *  *  *  *  *  *  *  *  *
*  H  *  *  *  *  *  *  *  *  *  *  *  *
*  H  *  *  *  *  *  *  *  *  *  *  *  *
*  H  *  *  *  *  *  *  *  *  *  *  *  *
*  H  *  *  *  *  *  *  *  *  *  *  *  *
*  H  *  *  *  *  *  *  *  *  *  *  *  *
*  H  *  *  *  *  *  *  *  *  *  *  *  *
*  H  *  *  *  *  *  *  *  *  *  *  *  *
*  H  *  *  *  *  *  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *  *  *  *  *  *
```

```
Enter draw command (enter Q to quit) H(2,4,4)Y



*  *  *  *  *  *  *  *  *  *  *  *  *  *
*  H  *  *  *  *  *  *  *  *  *  *  *  *
*  H  *  *  Y  Y  Y  Y  *  *  *  *  *  *
*  H  *  *  *  *  *  *  *  *  *  *  *  *
*  H  *  *  *  *  *  *  *  *  *  *  *  *
*  H  *  *  *  *  *  *  *  *  *  *  *  *
*  H  *  *  *  *  *  *  *  *  *  *  *  *
*  H  *  *  *  *  *  *  *  *  *  *  *  *
*  H  *  *  *  *  *  *  *  *  *  *  *  *
*  H  *  *  *  *  *  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *  *  *  *  *  *

Enter draw command (enter Q to quit) p(13,13,9999)@



*  *  *  *  *  *  *  *  *  *  *  *  *  *
*  H  *  *  *  *  *  *  *  *  *  *  *  *
*  H  *  *  Y  Y  Y  Y  *  *  *  *  *  *
*  H  *  *  *  *  *  *  *  *  *  *  *  *
*  H  *  *  *  *  *  *  *  *  *  *  *  *
*  H  *  *  *  *  *  *  *  *  *  *  *  *
*  H  *  *  *  *  *  *  *  *  *  *  *  *
*  H  *  *  *  *  *  *  *  *  *  *  *  *
*  H  *  *  *  *  *  *  *  *  *  *  *  *
*  H  *  *  *  *  *  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *  *  *  *  *  @
```

```
Enter draw command (enter Q to quit) q

student@maverick:/media/sf_VM/CSE1320/CA4$
```

Running program with invalid inputs (out of bounds)

```
How big is the array? (Enter a value between 1 and 20) 5
What is the background character? .

Draw commands start with

        P for a single point
        H for a horizontal line
        V for a vertical line

After the P/V/H, enter a row,col coordinate and the number of spots to mark
enclosed in () and separated by commas and then the character for the mark.
'X' will be used if a mark is not entered.  For example,

        P(2,3,1)*       start at point 2,3 in the array and mark one spot
                        with an *.  For P, the 3rd parameter is ignored.

        H(1,2,3)$       start at point 1,2 in the array and mark the next
                        3 spots to the right with $

        V(4,6,7)#       start at point 4,6 in the array and mark the next
                        7 spots down from the current position with #

Coordinates out of range and lines drawn past the borders are not allowed.

Enter Q at the draw command prompt to quit

Press <ENTER> to continue
```

```
.  .  .  .  .
.  .  .  .  .
.  .  .  .  .
.  .  .  .  .
.  .  .  .  .

Enter draw command (enter Q to quit) p(0,6,123)X

That draw command is out of range


.  .  .  .  .
.  .  .  .  .
.  .  .  .  .
.  .  .  .  .
.  .  .  .  .

Enter draw command (enter Q to quit) p(0,5,123)X

That draw command is out of range


.  .  .  .  .
.  .  .  .  .
.  .  .  .  .
.  .  .  .  .
.  .  .  .  .

Enter draw command (enter Q to quit) p(0,4,123)x


.  .  .  .  x
.  .  .  .  .
.  .  .  .  .
.  .  .  .  .
```

```
.  .  .  .  .

Enter draw command (enter Q to quit) h(2,2,5)z

That draw command is out of range



.  .  .  .  x
.  .  .  .  .
.  .  .  .  .
.  .  .  .  .
.  .  .  .  .

Enter draw command (enter Q to quit) H(2,2,2)Z



.  .  .  .  x
.  .  .  .  .
.  .  Z  Z  .
.  .  .  .  .
.  .  .  .  .

Enter draw command (enter Q to quit) V(-1,1,2)*

That draw command is out of range



.  .  .  .  x
.  .  .  .  .
.  .  Z  Z  .
.  .  .  .  .
.  .  .  .  .

Enter draw command (enter Q to quit) V(1,1,4)*
```

```
.  .  .  .  x
.  *  .  .  .
.  *  Z  Z  .
.  *  .  .  .
.  *  .  .  .

Enter draw command (enter Q to quit) V(1,0,5)*

That draw command is out of range



.  .  .  .  x
.  *  .  .  .
.  *  Z  Z  .
.  *  .  .  .
.  *  .  .  .

Enter draw command (enter Q to quit) Q

student@maverick:/media/sf_VM/CSE1320/CA4$
```

Checking array size

```
student@maverick:/media/sf_VM/CSE1320/CA4$ ./Code4_1000074079.e
How big is the array? (Enter a value between 1 and 20) 21
That value is outside of the max bounds of the array.  Please reenter

How big is the array? (Enter a value between 1 and 20) -1
That value is outside of the max bounds of the array.  Please reenter

How big is the array? (Enter a value between 1 and 20) 0
That value is outside of the max bounds of the array.  Please reenter

How big is the array? (Enter a value between 1 and 20) 4
```

Quitting at a draw command prompt

```
student@maverick:/media/sf_VM/CSE1320/CA4$ ./Code4_1000074079.e
How big is the array? (Enter a value between 1 and 20) 12
What is the background character? ^

Draw commands start with

        P for a single point
        H for a horizontal line
        V for a vertical line

After the P/V/H, enter a row,col coordinate and the number of spots to mark
enclosed in () and separated by commas and then the character for the mark.
'X' will be used if a mark is not entered.  For example,

        P(2,3,1)*       start at point 2,3 in the array and mark one spot
                        with an *.  For P, the 3rd parameter is ignored.

        H(1,2,3)$       start at point 1,2 in the array and mark the next
                        3 spots to the right with $

        V(4,6,7)#       start at point 4,6 in the array and mark the next
                        7 spots down from the current position with #

Coordinates out of range and lines drawn past the borders are not allowed.

Enter Q at the draw command prompt to quit

Press <ENTER> to continue


^  ^  ^  ^  ^  ^  ^  ^  ^  ^  ^  ^
^  ^  ^  ^  ^  ^  ^  ^  ^  ^  ^  ^
^  ^  ^  ^  ^  ^  ^  ^  ^  ^  ^  ^
^  ^  ^  ^  ^  ^  ^  ^  ^  ^  ^  ^
```

```
^  ^  ^  ^  ^  ^  ^  ^  ^  ^  ^  ^
^  ^  ^  ^  ^  ^  ^  ^  ^  ^  ^  ^
^  ^  ^  ^  ^  ^  ^  ^  ^  ^  ^  ^
^  ^  ^  ^  ^  ^  ^  ^  ^  ^  ^  ^
^  ^  ^  ^  ^  ^  ^  ^  ^  ^  ^  ^
^  ^  ^  ^  ^  ^  ^  ^  ^  ^  ^  ^
^  ^  ^  ^  ^  ^  ^  ^  ^  ^  ^  ^
^  ^  ^  ^  ^  ^  ^  ^  ^  ^  ^  ^


Enter draw command (enter Q to quit) q


student@maverick:/media/sf_VM/CSE1320/CA4$
```

Hints

The logic of parsing and storing the draw command entered at the prompt....

The function `strtok()` will give you a pointer to each token it finds between your delimiters so if the draw command is

```
H(1,2,3)%
```

then the first call to `strtok()` will give you a pointer to 'H'. You will then need to use `strcpy()` to store that H in a variable.

The next call to `strtok()` (using NULL instead of the array so it does not start over) will give you a pointer to '1'. You will then need to use `strcpy()` to store that character '1' in a variable. You can then use `atoi()` to convert your character '1' into an integer 1 which you will store in an integer variable.

You repeat this process for '2' and '3' including the `atoi()` step to get integer versions.

The final call to `strtok()` will get the '%'.

You will have 5 calls to `strtok()` total to parse one draw command. You should be using multiple `strtok()` statements to get each piece of the draw command. This is NOT a situation where you can use a `while` loop with `strtok()`.

Make sure you set your delimiter string in ALL of your `strtok()` statements to be "(),". Make `strtok()` work for you - don't do its job - give it all 3 delimiters and make `strtok()` figure it out. Don't give it one delimiter at a time.

Be sure to include `<stdlib.h>` when you want to use `atoi()`.

Remember the `strcpy()` copies into a string - a null terminated array.

If you try to copy into a `char` variable, then there won't be room for the null terminator so `strcpy()` will not compile

Be sure to give `strcpy()` a big enough array to store the H, P or V (for example) AND the null terminator.

Make your arrays big enough to hold possible data AND the null terminator. Then, if you need to refer to the first character, use `yourarrayname[0]` to access the character you stored. Or, pass the entire array to `atoi()` to convert.

HINT : you can actually just give `atoi()` the token pointer from `strtok()` and directly convert to an integer. It is not necessary to store the character version and then give that to `atoi()`.

As far as validation, for this program, we are assuming that the user enters the correct format. The only validation is making sure you have a P, V, H or Q after you parse the draw command.

If the user enters

```
H)1,2,3(&
```

then you should still use it because `strtok()` will still tokenize that correctly.

If the user enters

```
E(1,2,3)@
```

then that will be rejected as an invalid command because of the E.

When a draw command is entered, remember what `fgets()` is putting in the array.

If `H(1,2,3)&` is entered at the prompt, then `fgets()` will put

```
H(1,2,3)&\n
```

in the array.

If `H(1,2,3)` is entered at the prompt, then `fgets()` will put

```
H(1,2,3)\n
```

in the array.

So, for your fifth `strtok()` will either get the `&` or the `\n` (if the `&` mark is left off).

If your token is a `\n`, then you need to replace it with the default '`X`'.

Make sure the array you give `fgets()` is big enough - use 20 to make sure you have room for double digit numbers. Using too small of an array will cause issues with `strtok()`.

Remember that MAXMAPSIZE will be altered during grading. You should NOT be using it for any other purpose - it should only be used to define the 2D array and when referring to the dimensions of the 2D array.

Some logic to get you started

prompt for draw command

fgets drawcommandinput into an array that is at least 20 and does not use MAXMAPSIZE

strtok to get draw command (Q,H,P,V)

uppercase it

while (drawcommand is not Q)

{

    strtok the rest of the drawcommandinput

    draw the line/point

    prompt for draw command

    fgets drawcommandinput into array that is at least 20 and does not use MAXMAPSIZE

    strtok to get draw command (Q,H,P,V)

    uppercase it

}

If you are replacing the newline at the end of the draw command with \0 and then checking if mark is missing by looking for NULL, then you make encounter a seg fault when running the redirect file.

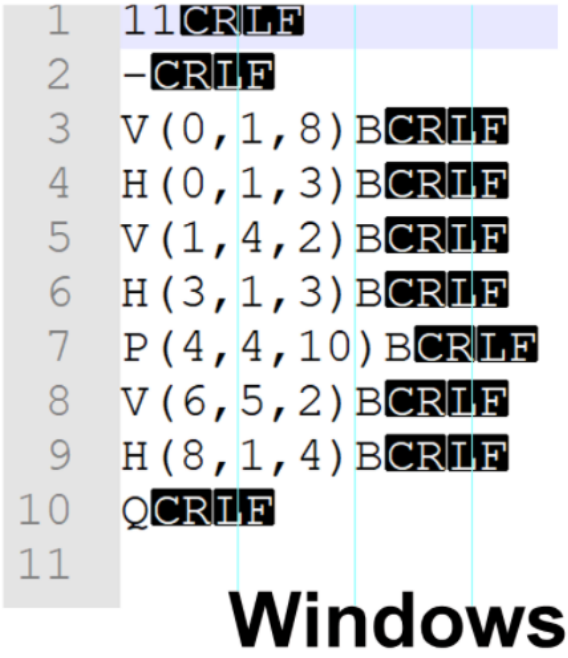
If the last line of your file is

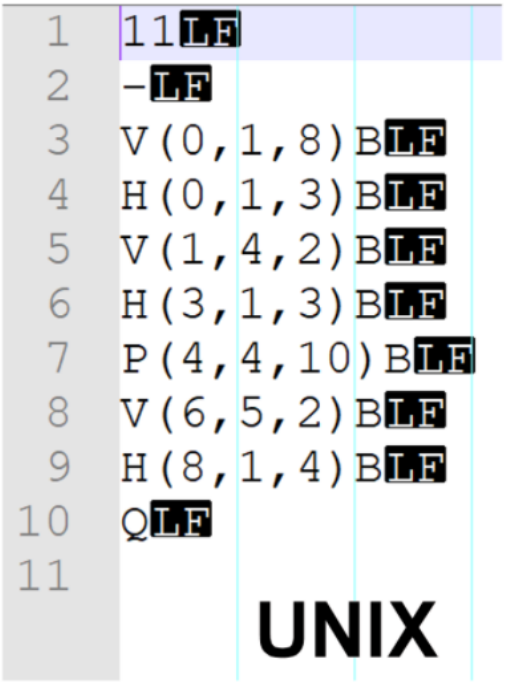
Q

and you don't have a blank line after it, then your fgets() will get the "Q" from the file and replace it with NULL and cause your program to seg fault.

You are better off to leave the \n on the end, detect if mark is missing because it has a value of \n and then you won't have the final Q problem at all.

UNIX Line Feeds

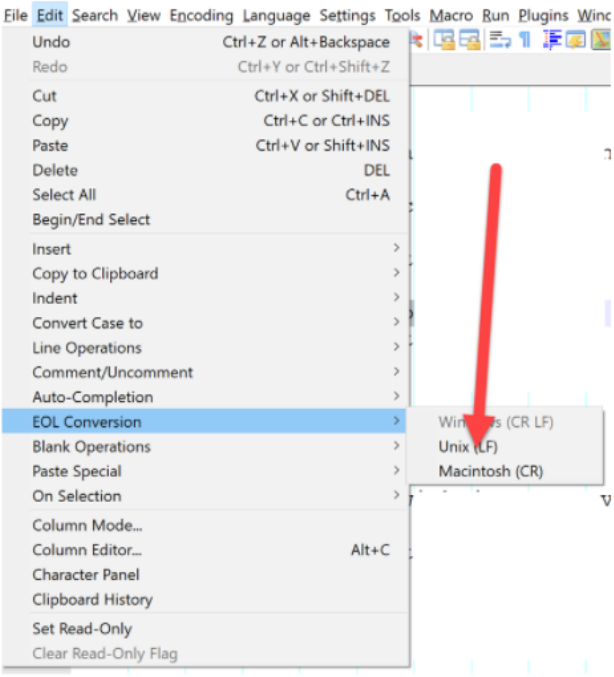




Windows files have CRLF and MAC files have CR. UNIX files need LF

Either figure out if your editor can change the End Of Line conversions (EOLs). Notepad++ can do it or you can run a UNIX command to change it.

UNIX Line Feeds




```
sed -i.old 's/\r$//' input.txt
```