CSCB09 2025 Summer Assignment 1

Due: June 8 11:59PM
This assignment is worth 10% of the course grade.

In this assignment, you will implement an interesting shell script and learn/use some Unix utility programs.

As usual, you should aim for reasonably efficient algorithms and reasonably organized, comprehensible code.

Correctness (mostly auto-testing) is worth 90% of the marks; code quality is worth 10%.

Editing A List in An Environment Variable

Many systems and programs designate environment variables to store lists of strings; usually the strings are separated by colons, but some systems use semicolons or other punctuations. Examples:

variable	example value
PATH	`/usr/local/bin:/usr/bin:/bin:/snap/bin`
TEXINPUTS	`.:aux:`
LS_COLORS	`rs=0:di=01;34:ln=01;36:mh=00` (etc)
PATH (Windows)	`C:\Windows;C:\Program Files (x86)\Steam`

They are very annoying to use! The values are hard to read because everything is cramped into one long line. And they are difficult to manually change—while appending and prepending are not too bad (but not great), deleting something in the middle is really tedious.

Let's write a shell function to help with that! (Quiz: Why would a shell script be futile?) The function is called `editlist` with this syntax:

```
editlist OPTION... [--] [STRING...]
```

option	meaning
`-l`	print current value, one element per line (see examples)
`-a`	append every STRING
`-p`	prepend every STRING (see examples for order)
`-d`	delete all occurrences of every STRING
`-e` VAR	the environment variable (default PATH)
`-s` SEP	the separator (default colon)

Exactly one of `-l`, `-a`, `-p`, `-d` is expected. If none or different ones are given, it is a user error: Exit with code 1; error messages are optional but should go to stderr only. If the same one is given more than once, it is up to you what to do.

If `-e` occurs multiple times, it is up to you which one takes effect. Similarly for `-s`.

The separator SEP, if given, is expected to be exactly one character. If not, it is up to you what to do.

If an unrecognized option is given, exit with code 1; error messages are optional but should go to stderr only.

The user may use `--` to signify the end of options—all subsequent arguments are considered STRINGs, even if they look like `-a` for example. If you use `getopts`, you get this feature for free.

The STRINGs are expected to not contain the separator. If a string contains the separator, it is up to you what to do.

Please define your `editlist` function in def-editlist.sh and hand it in. You may add helper functions.

Examples: The provided script example.sh loads the function definition and runs sample tests; test it with `sh example.sh`. The provided file expected.txt contains the exact expected output.

(You do know how to use pipelining/redirection and `diff` to check your output against expected.txt, right? Automarking will not accept any "invisible" difference in blanks and unprinted bytes. B09 is one of the few courses where you develop this sensitivity, as all programmers are supposed to!)

Tips:

- If the built-in `echo` command is messing up the \ character, find something else to use. Some candidates: `/bin/echo`, `printenv`.
- `grep` can help you delete, but you need to find and use a lot of non-default options.
- At some point you will need the built-in `eval` command because you are setting a variable, but its name cannot be hardcoded.

This assignment is designed to be easily solvable without creating intermediate "temp" files. Tests will be run in a docker container under heavy lockdown.