Command line arguments

sending in data

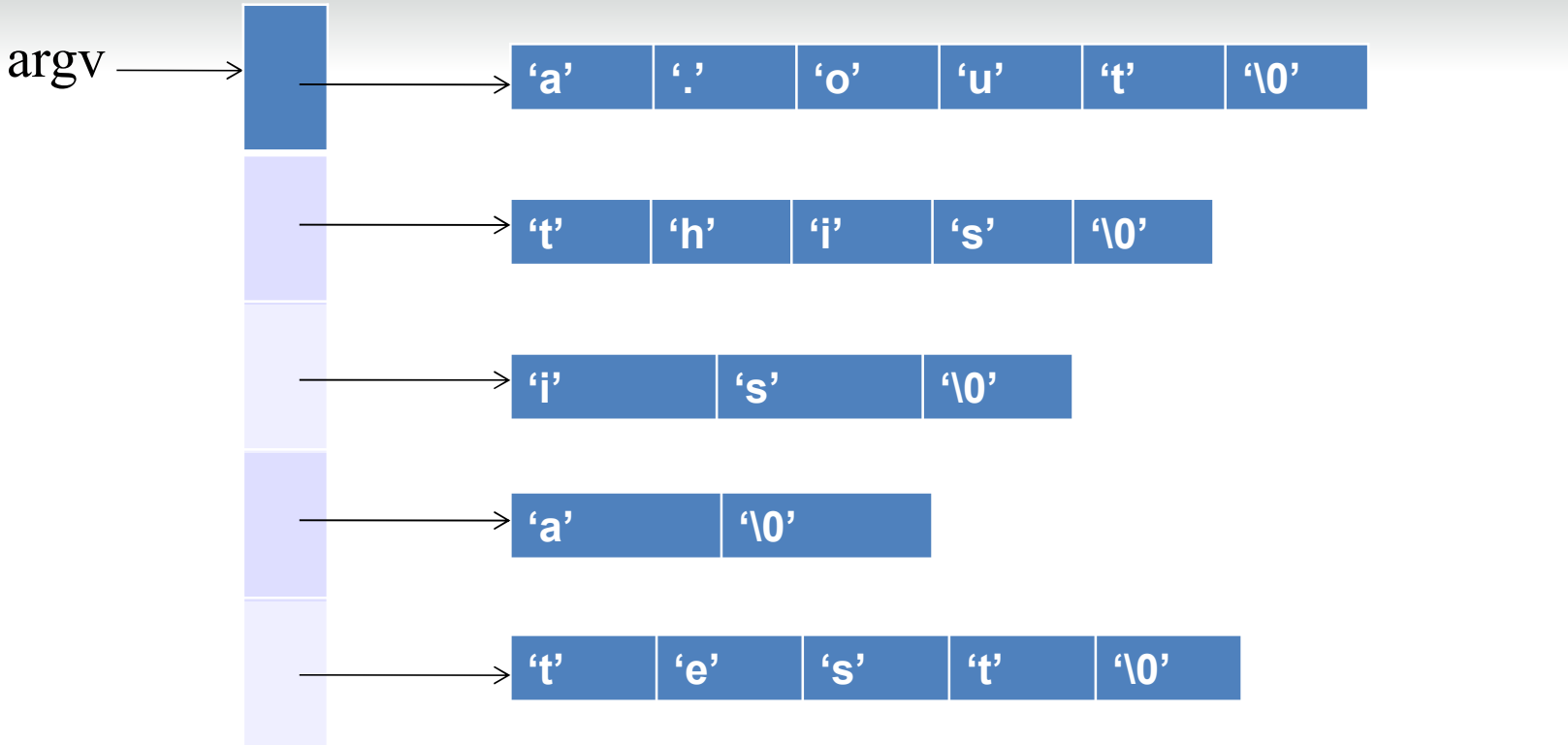
Good if we could send in data from the command line:

- provide data on the command line that the program could process
- Allow us to change how we run the program
- imitate Unix commands

`main(int argc, char **argv)`

- can now add parameters to main
- **`argc`** is a count of how many command args there were (including the program name)
- what the heck is char **`**argv`**?

pointer to pointer



Typed on the command line: `a.out this is a test`

argv

- `argv` points to an array, and each element of that array points to another array, an array of chars (a string)
- we can reference `argv[0]`, which is a pointer to a char array, a string, and process it

atoi and the like

You can convert a string to a number using `atoi, atol, atof, atod` (`int, long, float, double`) assuming that the string is indeed a number.

You must

```
#include<cstdlib>
```

Things brought in from C are automatically in the main namespace (no need for `std::`)

echoCommandLine

an echo program

names don't matter

- the variables can be named anything you like, but `argc` and `argv` are traditional
- `argc` is always at least 1 (the name of the program counts) and `argv[0]` points to the name of the program

check usage

- You can check for proper usage of the command line and report back with the proper call if the user does it wrong.

More complicated example

CLcheck