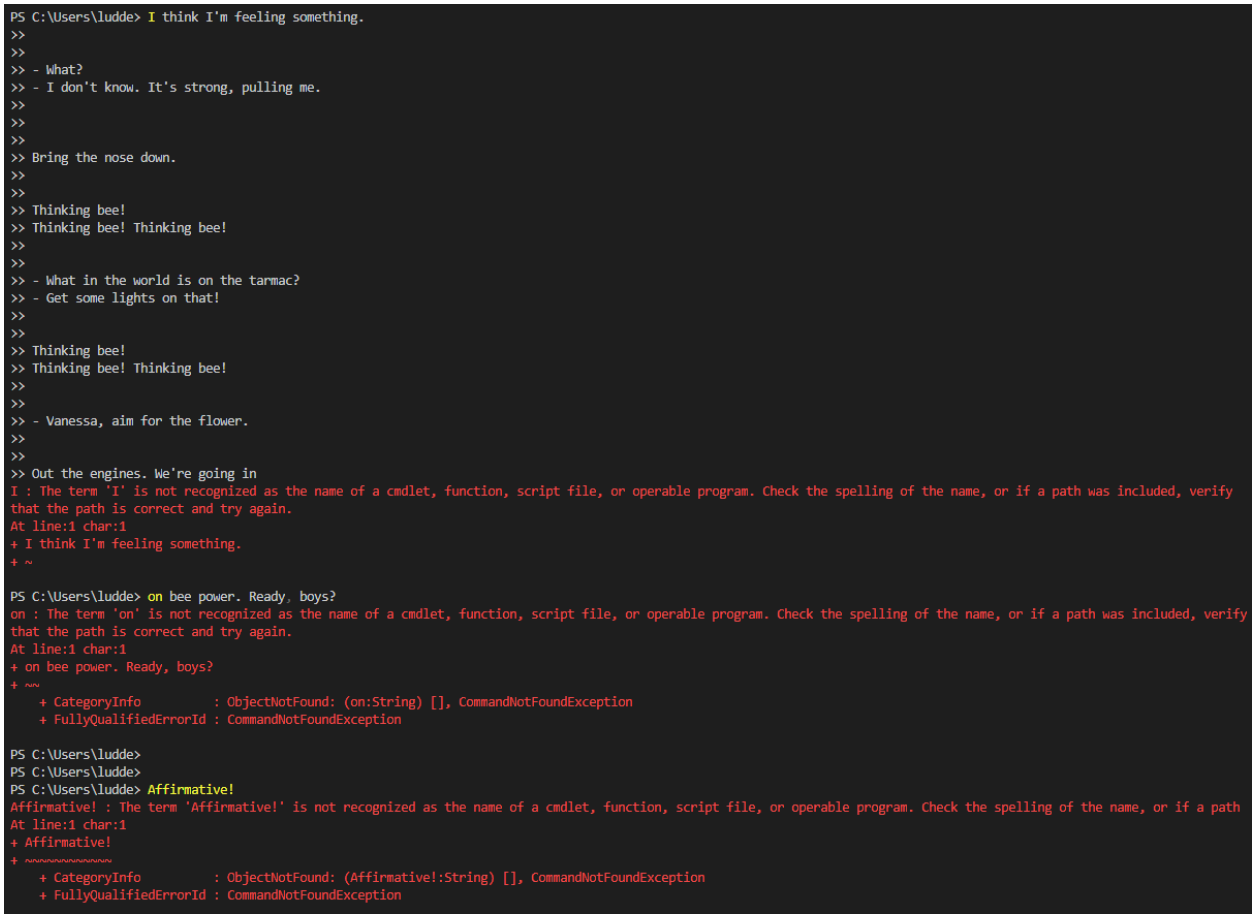

With a long input the characters will overflow the program and be read through the command line

13/10/20
Functions and Control Flow
https://secure.ecs.soton.ac.uk/notes/ellabs/1/c2/c2.pdf
https://www.arduino.cc/reference/en/language/variables/data-types/unsignedlong/
https://en.wikipedia.org/wiki/C_data_types
https://en.wikipedia.org/wiki/Printf_format_string
https://stackoverflow.com/questions/20186809/endless-loop-in-c-c

An unsigned long variable can be declared like so:

Syntax

```
unsigned long var = val;
```

Parameters

`var`: variable name.
`val`: the value you assign to that variable.

To initialize an unsigned long variable I have to make sure the compiler knows what data type I am using. When I declare my variable I must put the type at the end of the initialized value.

Unsigned long var = 10UL;

The format specifier for an unsigned long variable is %lu.

Fixed width data types are data types that have a specified length, because every system has its own default length This helps the portability and reliability of the code. To use a fixed width data type I must include the <stdint.h> library, and if using macros the <inttype.h> library.

When declaring the variable and wanting a fixed character width value I must use the library's data types:

Type category	Signed types			Unsigned types		
	Type	Minimum value	Maximum value	Type	Minimum value	Maximum value
Exact width	`intn_t`	`INTn_MIN`	`INTn_MAX`	`uintn_t`	0	`UINTn_MAX`
Least width	`int_leastn_t`	`INT_LEASTn_MIN`	`INT_LEASTn_MAX`	`uint_leastn_t`	0	`UINT_LEASTn_MAX`
Fastest	`int_fastn_t`	`INT_FASTn_MIN`	`INT_FASTn_MAX`	`uint_fastn_t`	0	`UINT_FASTn_MAX`
Pointer	`intptr_t`	`INTPTR_MIN`	`INTPTR_MAX`	`uintptr_t`	0	`UINTPTR_MAX`
Maximum width	`intmax_t`	`INTMAX_MIN`	`INTMAX_MAX`	`uintmax_t`	0	`UINTMAX_MAX`

n is the fixed width of the data type, for example

int_fast8_t

Is the fastest data type of length 8.

Within the printf() and scanf() format strings the fixed width format specifier is %d.

To create an endless loop I can use the for or while commands. I can use the for command to create an endless loop like so:

```
for (;;) {
...
}
```

> is an "infinite" loop, presumably to be broken by other means, such as a break or return. Whether to use while or for is largely a matter of personal preference.

Using the while command to create the endless loop can be done like so:

while(1){}

However this is likely to be misidentified as an error by the compiler because the statement is always true.

To create a sine function I can sin() function in the <math.h> library. This gives the radian sine value of an input. To increase the frequency I can give the input a coefficient.

The sine value can be scaled to the range of 0 to 1 using the formula y=(sin(x)+1)/2

My version of the plotval function:

```c
/*plotval.*/

/*libraries*/
#include <stdio.h>
#include <stdint.h>
#include <math.h>

/*definitions*/
#define MAX_TERMINAL 120

/*plotval function*/
int plotval(float ploti)                                    /*Declare plotval function and plot input*/
{
    for (int i = 0; i <= roundf(ploti * MAX_TERMINAL); i++)     /*Loop till at proportion of the screen to sin value*/
    {
        printf(" ");
    }
    printf("*\n");                                          /*Mark with an asterix*/
}
```