

Validating strings

The standard library header file named `ctype.h` contains a number of functions that are useful to perform tests on characters. To make these available to a program, the `ctype.h` header file must be added with an `#include` directive at the start of the program.

Within the `ctype.h` header file, the `isalpha()` function tests whether a character is alphabetical, whereas `isdigit()` tests whether a character is numerical. Similarly, the `ispunct()` function tests whether a character is any other printable symbol, whereas the `isspace()` function tests whether a “character” is in fact a space.

Additionally, the `ctype.h` header provides `isupper()` and `islower()` functions to test the character case, along with `toupper()` and `tolower()` functions to change the character case.

Each test function returns a non-zero value (not necessarily a 1) when the tested character is as expected, but always returns a zero if it is not as expected.

These testing functions can be used to validate user string input by looping through characters in the string to examine each in turn. Where a character does not meet a required condition, a “flag” variable can be set to advise the user that the entry is invalid.



isval.c

1

Begin a new program with a preprocessor instruction to include the standard input/output and string functions

```
#include <stdio.h>
#include <ctype.h>
```

2

Add a main function that declares three variables, initializing a flag variable with a numeric true value (1)

```
int main()
{
    char str[7];
    int i;
    int flag = 1;
}
```

3

Next, in the main function, request user input then assign that input as a string to the array variable

```
puts( "Enter six digits without any spaces..." );
gets( str );
```