

## Documentation for Utility.h and Utility.c

Steven Andrews, © 2003

See the document "LibDoc" for general information about this and other libraries.

```
#define inputf(E1,E2,A) {printf(E1,A);scanf(E2,&##A);}
#define lcase(A) ((A)>='A'&&(A)<='Z'?((A)-'A'+'a'):(A))
#define isnum(A) ((A)>='0'&&(A)<='9' || (A)=='-')

void WaitTime(float dt);
char Wait4Event();
char Wait4Mouse(int *x,int *y);
char inkey();
void beep();
void SiouxWindow(float left,float top,int columns,int rows,int event);
int CheckEvents(EventRecord *eventptr);
```

Requires: <stdio.h>, <stdlib.h>, <string.h>

Example program: SpectFit.c

History: Written 1/99; moderate testing. Works with Metrowerks C. String routines were moved to newly created string.c library 11/01.

This library contains several useful low level routines for user interaction. Many of them are Macintosh specific. Added SIOUX (standard input/output exchange) stuff 3/25/02.

### Functions and definitions

`inputf` is similar to the Basic routine for inputting something. E1 is a string that is displayed, E2 is the scanf formatting string for the input, and A is the variable where the result should be stored. Example: `inputf("Enter number:", "%f", x);`. It hasn't turned out to be as useful as expected, so it may be deleted at some point.

`lcase` takes a character and, if it is upper case, returns the same letter in lower case. If it wasn't upper case, then the original is returned. This is nearly redundant with the standard library routine `tolower`, which should be used if possible.

`isnum` returns 1 if a character is a number or a '-' sign, and 0 otherwise. This is nearly redundant with the standard library routine `isdigit`, which should be used if possible.

`WaitTime` pauses execution for dt seconds.

`Wait4Event` runs an endless loop until a key is pressed, after which it returns the character.

`Wait4Mouse` runs an endless loop until the mouse button is pressed, at which point the mouse position is returned.

`inkey` works just like the function of the same name in Basic. It returns either 0 for no key pressed or the ascii code of the key that was pressed.

`beep` causes a beep.

`SiouxWindow` sets up the standard i/o window to appear somewhere other than the default location. `left` and `top` are values that are fractions of the screen dimensions (as for `MakeWindow` in `Plot.c`). The new window is sized for `columns` columns and `rows`

rows of text. If event is 0, event managing is taken care of automatically, whereas 1 turns off automatic event handling, and it has to be done explicitly. CheckEvents is just my version of the standard toolbox function GetNextEvent. It also calls the Sioux window to see if it wants the event. Otherwise it's sent back in event, and the return value is 1 if there's an event and 0 if not.