Input

Suppose you want to read in an arbitrary number of floats. How do you do this?

1. Just let the number of x's be an input parameter `n`:

```
#include <iostream.h>
int main()
{
 int n;
 float x;
 cout << "Number of x's" << endl;
 cin >> n;
 int i;
 for(i=0; i<n; i++)
 {
   cout<<"Enter x" << endl;
   cin >> x;
 }
 return 0;
}
```

2. Use `control-D` to end the program:

```
#include <iostream.h>
int main()
{
 float x;
 while(cin>>x)
  //do whatever
 return 0;
}
```

3. Have a character at the end of the stream, like "q". Then use

```
#include <iostream.h>
int main()
{
  char c;
  float x;
  while (1)
  {
    cin >> c;
    if(c == 'q')
      {exit(1);}
    else
      {
      cin.putback(c);              //put c back in the input stream
      cin >> x;
      }
  }
   return 0;
 }
```

4. You can use argc and argv if you enter everything on the command line of your program. argc is the number of items listed in the command line and argv is an array of those items where each item is regarded as a string.

```
#include <stdlib.h>
int main(int argc, char **argv)
{
   int i;
   float x;
   for(i=1; i<argc; i++)          //start at i=1 because the executable
                                  //is not a number
    {
     x=atof(argv[i]);             //atof turns ascii into float
    }
   return 0;
}
```

The way you run this is:

```
moses% a.out 24.3 534.5 664.3
```

Here `argc = 4` and `argv[0]` is `a.out`.

Input and Output Files

The quick and dirty way to read from one file and write to another file is to use `cout` and `cin` as before, but on the command line type

```
moses% a.out < input.dat > output.dat
```

where `input.dat` is the input file and `output.dat` is the output file. You can use any name for the input and output files. The drawback is that you can have only one input file and one output file.

A better way to open files to read from and write to:

```
#include <fstream.h>

int main() {
 ifstream infile("input.dat");     //input.dat is the name of the file
                                   //in your directory
 ofstream outfile("output.dat");   //output.dat is the name of the file
                                   //that will be created in your directory
 float x;
   while(infile >> x)              //detects end-of-file and exits loop
      { outfile << "x = " << x << endl; }
 infile.close();
 outfile.close();
 return 0;
}
```

There is nothing sacred about the words `infile` and `outfile`. You can use anyword in their place. Notice that `infile` acts like `cin` and `outfile` acts like `cout`.

We can open a file so that anything written to it is appended to the end:

```
#include <fstream.h>

int main() {
 ifstream infile("input.dat");
 ofstream outfile("output.dat",ios::app);  //append to output.dat
 float x;
   while(infile >> x) {
       outfile << "x = " << x << endl; }
 infile.close();
 outfile.close();
 return 0;
}
```

It is also possible to open a file for both input and output. For example,

```
#include <fstream.h>

 int main() {
  fstream inout("input.dat",ios::in|ios::out);
  float x;
    inout >> x;
    inout<< endl << "x = " << x << endl;
  inout.close();
  return 0;
 }
```