Bring it In – Getting Data Into a Program

Now that you have mastered the basics of output, storage and processing, it is time to explore the final, main task of programming, which is input. When working with the console interface, the most commonly used command will be the scanf(), which is similar to the printf() in how it handles information. The ‘f’ in each of these indicates that for both commands, a **format specifier** indicates the type (or format) of the data being handled.

Example of the use of scanf()

```
#include <stdio.h>

int main(void)
{
    int age;                                            //Stores an age in years

    printf(“\nPlease enter your age in years, and press return è”);
    scanf(“%i”,&age);                                   //Note the &!!!

    printf(“Really?? You don’t look %i years old!!\n\n”,age);

    return 0;
}
```

The most important difference in the use of scanf() is that it must be ‘told’ where to store the number, which is the job of the & (ampersand). The & is a pointer that indicates the address, or location in memory, where the value is to be saved. For all numeric input, the ampersand MUST be used. The format specifiers used in scanf() are the same as those for printf(), i.e. %i,%d,%f,%c,%G and several others may be used.

Exercise

Write a properly structured C program that will allow a user to enter personal information in both number (integer and float) and character form. Store all of this data as appropriate variables, and then redisplay the information all together in the console.