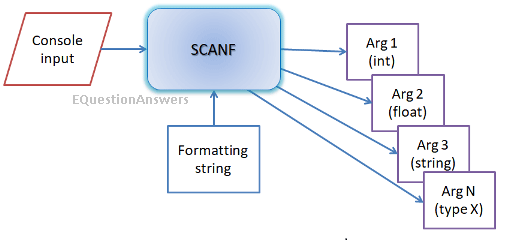
**Scanf working principle**

scanf or scan function in C takes a formatting string as input and couple of optional variables as output reference arguments. It converts the scanned string to variables and copies to the output variables.



Scanf is reverse process of printf. Scanf reads console input string. It iterates each characters of user provided string and stops at "%". Now scanf reads a line from stdin. User's input comes as a string. It converts string to char, int, long, float, double and sets the value of the pointer located at the argument. In care of string it simply copies the string to the output

**Implement scanf**

Let us implement our own scan function. This is only for the understanding purpose. We name it scan(). It has one string argument (str) and rest are variable arguments. Variable arguments are managed by macros like va\_start, va\_arg and va\_end. A temporary buffer (buff) is there to copy buffer from stdin. In a while loop we read characters from stdin and and copy to buff. Loop terminates when user presses return key. Now we have another while loop to scan the buffer. We iterate character by character and we check for "%". Once we found it, we check the next character. This is the formatting character. Formatting character says how to argument is formatted in the buffer. Scanf supports varieties of formatting. C is for character, d for decimal integer, f for floating point, x for hexadecimal and s for strings. We pick the argument variable using va\_arg(). Argument variable is then converted from string to desired type. character can be copied as it is and strtol() function is used for [string to integer](http://equestionanswers.com/c/c-string-to-int.php) conversion.

C strtol() function has been used to convert argument [string to integer](http://equestionanswers.com/c/c-string-to-int.php). String to integer conversion is a process to take each ASCII characters and convert those to integer. Visit our topic [string to integer conversion](http://equestionanswers.com/c/c-string-to-int.php)for further understanding.

This process of taking each character and convert formatted string to integers repeats until the string is terminated to last NULL character. For simplicity we have implemented only c, d, x formatting cases. Now at the end we have the output string ready. Scanf then returns the number of output argument is successfully converted and exits.

/\* Note: this is a minimal scanf implementation \*/  
/\* This is for building understanding only \*/  
**#include** <stdio.h>  
**#include** <stdlib.h>  
**#include**<stdarg.h>  
**int** scan (**char** \* str, ...)  
{  
  va\_list vl;  
  **int** i = 0, j=0, ret = 0;  
  **char** buff[100] = {0}, tmp[20], c;  
  **char** \*out\_loc;  
  **while**(c != '\n') {  
    **if** (fread(&c, 1, 1, stdin)) {  
      buff[i] = c;  
      i++;  
    }  
  }  
  va\_start( vl, str );  
  i = 0;  
  **while** (str && str[i]) {  
    **if** (str[i] == '%') {  
      i++;  
    **switch** (str[i]) {  
        **case** 'c': {  
          \* (**char** \*)va\_arg( vl, **char**\* ) = buff[j];  
          j++;  
          ret ++;  
          break;  
        }  
        **case** 'd': {  
          \* (**int** \*)va\_arg( vl, **int**\* ) = \  
          strtol(&buff[j], &out\_loc, 10);  
          j += out\_loc -&buff[j];  
          ret++;  
          break;  
        }  
        **case** 'x': {  
          \* (**int** \*)va\_arg( vl, **int**\* ) = \  
          strtol(&buff[j], &out\_loc, 16);  
          j += out\_loc -&buff[j];  
          ret++;  
          break;  
        }  
        **case** 'o': {  
          \* (**int** \*)va\_arg( vl, **int**\* ) = \  
          strtol(&buff[j], &out\_loc, 8);  
          j += out\_loc -&buff[j];  
          ret++;  
          break;  
        }  
        **case** 's': {  
          out\_loc = (**char** \*)va\_arg( vl, **char**\* );  
          strcpy(out\_loc, &buff[j]);  
          j += strlen(&buff[j]);  
          ret++;  
          break;  
        }  
      }  
    } **else** {  
      buff[j] =str[i];  
      j++;  
    }  
    i++;  
  }  
  va\_end(vl);  
   **return** ret;  
}  
**int** main(**int** argc, **char** \*argv[])  
{  
  **char** c;  
  **int** i;  
  **int** h;  
  **int** o;  
  **char** str\_buff[20];  
  **int** ret = 0;  
  printf("Enter char int hex oct string\n");  
  ret = scan("%c %d %x %o %s", &c, &i, &h, &o, str\_buff);  
  printf("C = %c, I = %d, H = %d, O = %d, S = %s returns %d", c, i, h, o, str\_buff, ret);  
  **return** 0;  
}

**Output**

Enter char int hex oct string

A 10 100 1000 Hello from scanf!

C = A, I = 10, H = 256, O = 512, S = Hello from scanf!

returns 5

http://www.equestionanswers.com/c/c-printf-scanf-working-principle.php