Tutorial 4 - C Examples

This tutorial provides some examples of C code.

The first example example copies characters from one array to another but reverses the order of the characters. Note that when the values in arrays passed to functions are altered the corresponding values in the calling function are altered. This is not the case with scalar variables. Thus the value of the array passed as 'y' will change, but the variable (if any) used to pass the value of 'n' won't.

```
/* Reverse the order of the n characters
   in the character array (string) x and
   put the result in the array y.  */

void reverse ( char x[], char y[], int n )
{
  int i ;
  i = 0 ;
  while ( n > 0 ) {
    y[n-1] = x[i] ;
    i = i + 1 ;
    n = n - 1 ;
  }
}
```

The second example converts a string to upper case. Note the use of two character constants to express the difference between the start of the ASCII codes for the upper and lower case alphabets.

```
/* Convert a string to uppper case. Note
   that both upper- and lower-case
   letters appear in order in the ASCII
   table. */

void strtoupper ( char x[], int n )
{
  int i ;
  i = 0 ;
  while ( i < n ) {
    if ( x[i] >= 'a' && x[i] <= 'z' ) {
      x[i] = x[i] - ( 'a' - 'A' ) ;
    }
    i = i + 1 ;
  }
}
```

The next example shows how you can put more complex data structures into an array.

```
/* Increment a time value given as a
   three element array consisting of the
   hours (index=2), minutes (index=1)
   and seconds (index=0). */

void nextsecond ( int hms[] )
{
  hms[0] = hms[0] + 1 ;
  if ( hms[0] >= 60 ) {
    hms[0] = 0 ;
    hms[1] = hms[1] + 1 ;
    if ( hms[1] >= 60 ) {
      hms[1] = 0 ;
      hms[2] = hms[2] + 1 ;
      if ( hms[2] >= 24 ) {
        hms[2] = 0 ;
      }
    }
  }
}
```

The next example does the same thing but using a loop to iterate over the three elements of the array. In this case the code is more complex and harder to understand. But if the time structure had many elements then this algorithm might be simpler.

```
/* Another way of doing the same thing.
   Note syntax for initializing array
   values. */

void nextsecond2 ( int hms[] )
{
  int i, done, limit[3] = { 60, 60, 24 } ;
  i = done = 0 ;
  while ( ! done && i < 3 ) {
    hms[i] = hms[i] + 1 ;
    if ( hms[i] >= limit[i] ) {
      hms[i] = 0 ;
      if ( i < 2 ) {
        hms[i+1] = hms[i+1] + 1 ;
      }
      i = i + 1 ;
    } else {
      done = 1 ;
    }
  }
}
```

This example shows how the bitwise logical operators can be used to "pick out" individual bits in a

number and how division by a power of two shifts the bits in a number to right.

```
/* print the octal (base 8) representation of a number.  It is
   assumed the number can be represented as 5 or fewer octal
   digits. */

void prtoctal ( int n )
{
  int i, digits[5] ;

  /* find the octal digits in n from LS to MS */
  i = 0 ;
  while ( i < 5 ) {
    digits[i] = n & 0x07 ;      /* save the LS 3 bits */
    n = n / 8 ;                 /* shift 3 bits to the right */
    i = i + 1 ;
  }

  /* print the octal digits in the right order */
  i = 4 ;
  while ( i >= 0 ) {
    printf ( "%d", digits[i] ) ;
    i = i - 1 ;
  }
}
```

The last function demonstrates how the above functions could be called.

```
/* test the above functions */

void main ( void )
{
  char z[5] ;
  int time[3] = { 59, 59, 23 } ;
  reverse ( "abcde", z, 5 ) ;
  printf ( "abcde reversed is %s\n", z ) ;

  strtoupper ( z, 5 ) ;
  printf ( "converted to upper case is %s\n", z ) ;

  printf ( "start time is %d:%d:%d\n", time[2], time[1], time[0] ) ;
  nextsecond ( time ) ;
  printf ( "next second is %d:%d:%d\n", time[2], time[1], time[0] ) ;
  nextsecond2 ( time ) ;
  printf ( "next second is %d:%d:%d\n", time[2], time[1], time[0] ) ;

  prtoctal ( 12345 ) ;
}
```