```c
#define NULL  (void *)0

int from_ascii(int c, int b)
{
   int i;

   if(c >= '0' && c <= '9')
      i = c - '0';
   else if(c >= 'A' && c <= 'Z')
      i = c - 'A' + 10;
   else if(c >= 'a' && c <= 'z')
      i = c - 'a' + 10;
   else
      i = -1;
   if(i >= b)
      i = -1;
   return i;
}

int strtoi(char *s, int b, char **e)
{
   int sign, c, n = 0;

   // get sign
   sign = 1;
   if(*s == '+')
      s++;
   if(*s == '-')
   {
      s++;
      sign = -1;
   }
   // get base
   if(b == 0 && s[0] == '0')
   {
      c = (int)s[1];    // convert char to int (MIPS "LB" does this)
      if(c >= 'A' && c <= 'Z')
         c += 'a' -'A';
      if(c == 'b') b =  2; // 0b means base  2 (binary)
      if(c == 'o') b =  8; // 0o means base  8 (octal)
      if(c == 'd') b = 10; // 0d means base  2 (decimal)
      if(c == 'h') b = 16; // 0h means base 16 (hexadecimal)
      if(b != 0)
         s += 2;
   }
   if(b < 2 || b > 36)
      b = 2;

   while ((c = from_ascii((int)(*s++), b)) >= 0)
      n = b * n + c;

   // finish
   if(sign < 0)
      n = -n;
   if(e != NULL)
      *e = s;
   return n;
}


/* main() in the next page */
```

```c
/*****************************************************************************
 main() to test the strtoi() function
*****************************************************************************/
int main(void)
{
   char s[]="0b111010";

   print_string(s);
   print_string(" -> ");
   print_int10(strtoi(s, 0, NULL)); // should print 58
   print_string("\n");
   return 0;
}

/*****************************************************************************
 Alternative main() to test the strtoi() function
*****************************************************************************/
int main(int argc, char *argv[])
{
   int val;

   if(argc == 2)
      val = strtoi(argv[0], strtoi(argv[1], 10, NULL), NULL);
   else if(argc == 1)
      val = strtoi(argv[0], 0, NULL);
   else
   {
      print_string("Invalid input. Should be: <string> [base]\n");
      return -1;
   }
   print_string(argv[0]);
   print_string(" -> ");
   print_int10(val);
   print_string("\n");
   return 0;
}
```