59. An entire string is output by calling *print*. Note that if we are outputting the single standard ASCII character `c`, we could call *print*(`'c'`), since `'c'` ≡ 99 is the number of a single-character string, as explained above. But *print_char*(`'c'`) is quicker, so TEX goes directly to the *print_char* routine when it knows that this is safe. (The present implementation assumes that it is always safe to print a visible ASCII character.)

```
⟨ Basic printing procedures 57 ⟩ +≡
  void print(int s)    /* prints string s */
  { pool_pointer j;    /* current character code position */
    int nl;    /* new-line character to restore */
    if (s ≥ str_ptr) s = ⟨ "???" 1381 ⟩;    /* this can't happen */
    else if (s < 256)
      if (s < 0) s = ⟨ "???" 1381 ⟩;    /* can't happen */
      else { if (selector > pseudo) { print_char(s);
            return;    /* internal strings are not expanded */
          }
          if ((⟨ Character s is the current new-line character 244 ⟩))
            if (selector < pseudo) { print_ln( );
              return;
            }
          nl = new_line_char;
          new_line_char = −1;    /* temporarily disable new-line character */
          j = str_start[s];
          while (j < str_start[s + 1]) { print_char(so(str_pool[j]));
            incr(j);
          }
          new_line_char = nl;
          return;
        }
    j = str_start[s];
    while (j < str_start[s + 1]) { print_char(so(str_pool[j]));
      incr(j);
    }
  }
  void print_str(char *s)    /* the simple version */
  {
    while (*s ≠ 0) print_char(*s++); }
```

60. Control sequence names, file names, and strings constructed with `\string` might contain **ASCII_code** values that can't be printed using *print_char*. Therefore we use *slow_print* for them:

```
⟨ Basic printing procedures 57 ⟩ +≡
  void slow_print(int s)    /* prints string s */
  { pool_pointer j;    /* current character code position */
    if ((s ≥ str_ptr) ∨ (s < 256)) print(s);
    else { j = str_start[s];
      while (j < str_start[s + 1]) { print(so(str_pool[j]));
        incr(j);
      }
    }
  }
```