Imprimindo números de ponto flutuante

4. Postlúdio

A biblioteca C padrão nunca parece fazer exatamente o que você deseja para imprimir carros alegóricos. Se quiser notação científica, você pode usar "%e", mas 0 será impresso como 0,000000e+00. Ou você pode usar% f, mas números grandes produzem longas sequências de dígitos em vez da notação científica que você prefere.

Como presente de despedida, aqui está uma rotina que imprime números reais de maneira um pouco melhor, ajustando automaticamente os códigos de formato dependendo do tipo de número que você fornecer. Você pode especificar o quão grande ou pequeno um número pode ficar antes de passar para a notação científica e ainda pode especificar as larguras dos campos como no formato "%n.nf" usual.

```
#include <ieee754.h>
#define LOG2_10 3.321928095

#define flt_zero(x) (fabs(x) < EPSILON

int max_digs_rt = 3;  /* maximum # of
                         scientific no
int max_digs_lf = 5;  /* max # of digi

void print_real(double r, int width, i
{
    int mag;
    double fpart, temp;
    char format[8];
    char num_format[3] = {'l',0,0};
    union ieee754_double *dl;

    dl = (union ieee754_double*)&r;
    mag = (dl->ieee.exponent - IEEE754
    if (r == 0)
        mag = 0;
    if ((mag > max_digs_lf-1) || (mag
        num_format[1] = 'e';
        temp = r/pow(10, mag);       /*
        fpart = temp - floor(temp); /*
    }
    else {
        num_format[1] = 'f';
        fpart = r - floor(r);
    }
    if (flt_zero(fpart))
        dec = 0;
    if (width == 0) {
        snprintf(format, 8, "%%.%d%s",
    }
    else {
```

```
37              snprintf(format, 8, "%%%d.%d%s
38          }
39          printf(format, r);
40      }
```