

MAC 2166 – Introdução à Computação

POLI - PRIMEIRO SEMESTRE DE 2007

Material Didático

Prof. Ronaldo Fumio Hashimoto

UMA NOTA SOBRE / E %

Introdução

Nesta aula vamos mostrar alguns aspectos dos operadores / (divisão inteira) e % (resto de divisão inteira) para solução de problemas em MAC2166.

Divisão Inteira

Vimos na aula de comandos básicos, que o resultado de uma divisão inteira é sempre um número inteiro. Assim, o resultado de $3/4$ é **zero** (e não 0.75 , que é um número real) e o resultado de $9/2$ é 4 , $9/3$ é 3 , $9/4$ é 2 , e assim por diante. A parte fracionária é simplesmente eliminada (ou truncada ao invés de ser aproximada para um valor inteiro mais próximo).

Agora, o comentário desta aula é para o uso desta operação na resolução de alguns tipos de exercícios em MAC2166.

Existem exercícios em que há a necessidade de manipular os dígitos de um número inteiro. Para este tipo de exercício, é possível arrancar o último dígito de um número inteiro fazendo uma divisão inteira por 10. Por exemplo, $123/10=12$ (123 sem o último dígito). Assim, o comando $a = n / 10$ faz com que a variável a guarde o conteúdo da variável n sem o último dígito.

Resto de uma Divisão Inteira

Como já vimos em aulas anteriores, a operação $a \% b$ é o resto da divisão de a por b . Exemplos:

- $10 \% 4 = 2$.
- $3 \% 10 = 3$. Observe que 3 é menor que 10.
- $3 \% 2 = 1$.

Existem exercícios em que há a necessidade de se extrair os dígitos de um número inteiro positivo. Para este tipo de exercício, é possível obter o último dígito de um número inteiro calculando o resto de sua divisão inteira por 10. Por exemplo, $123\%10=3$ (último dígito de 123). Assim, o comando $a = n \% 10$ faz com que a variável a guarde o último dígito do número guardado na variável n .

Outros exercícios podem exigir a verificação se um número inteiro positivo n é par ou ímpar. Neste caso, basta verificar se o resto da divisão de n por 2 é 0 (zero) ou 1 (um), respectivamente.

Outros exercícios podem exigir a verificação se um número inteiro n é divisível ou não por um outro inteiro b (diferente de zero). Neste caso, basta verificar se o resto da divisão de n por b é 0 (zero) ou não, respectivamente.

Um Exemplo

Como um exemplo, vamos resolver o seguinte exercício:

Dados um número inteiro $n > 0$, e um dígito d ($0 \leq d \leq 9$), determinar o quantas vezes d ocorre em n . Por exemplo, 3 ocorre 2 vezes em 63453.

A seqüência a ser gerada aqui é composta pelos dígitos de n . Para cada dígito b de n , verificar se b é igual a d . Em caso afirmativo, aumentar um contador de um.

Para obter um dígito b de n , vamos fazer calcular o resto da divisão de n por 10, ou seja, $b = n \% 10$.

Para gerar a seqüência composta pelos dígitos de n , vamos usar então a seguinte repetição:

```
1     while (n>0) {
2         b = n % 10; /* último dígito de n */
3         n = n / 10; /* arranca último dígito de n */
4     }
```

Na linha 2, a variável b recebe o último dígito de n . Na linha 3, atualizamos a variável n para guardar seu conteúdo antigo sem o último dígito, para que na próxima vez, a variável b fique com um outro dígito de n . A condição do `while` garante que quando n for igual a zero, a seqüência de todos os dígitos de n já foi gerada.

Assim, para terminar, basta colocar um contador e um `if` para verificar se b é igual a d :

```
1     printf ("Entre com n>0: ");
2     scanf ("%d", &n);
3     printf ("Entre com d (0<=d<=9): ");
4     scanf ("%d", &d);
5     count = 0;
6     while (n>0) {
7         b = n % 10; /* último dígito de n */
8         if (b == d) {
9             count = count + 1;
10        }
11        n = n / 10; /* arranca último dígito de n */
12    }
13    printf ("%d ocorre %d vezes em %d\n", d, count, n);
```

Tem um erro aí em cima! Você saberia detectar? Sempre quando o fluxo do programa sai do laço, o valor da variável n é igual a zero! Pois, caso contrário, o fluxo do programa ainda estaria dentro do laço. Neste caso, o último `printf` sempre vai imprimir valor zero para a variável n . Como corrigir isso?

Exercícios que Usam estes Conceitos

1. Dado um inteiro $n > 0$, calcular a soma dos dígitos de n .
2. Dado um número natural na base binária, transformá-lo para a base decimal.

Exemplo: Dado 10010 a saída será 18, pois

$$1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 18.$$

3. Dado um número natural na base decimal, transformá-lo para a base binária.

Exemplo: Dado 18 a saída deverá ser 10010.

4. Qualquer número natural de quatro algarismos pode ser dividido em duas dezenas formadas pelos seus dois primeiros e dois últimos dígitos.

Exemplos:

- 1297: 12 e 97.
- 5314: 53 e 14.

Escreva um programa que imprime todos os milhares (4 algarismos) cuja raiz quadrada seja a soma das dezenas formadas pela divisão acima.

Exemplo: raiz de 9801 = 99 = 98 + 01.

Portanto 9801 é um dos números a ser impresso.

5. Dizemos que um número natural n é palíndromo se

- (a) o primeiro algarismo de n é igual ao seu último algarismo,
- (b) o segundo algarismo de n é igual ao penúltimo algarismo,
- (c) e assim sucessivamente.

Exemplos:

- 567765 e 32423 são palíndromos.
- 567675 não é palíndromo.

Dado um número natural $n > 10$, verificar se n é palíndromo.

6. Dados $n > 0$ e uma seqüência de n números inteiros, determinar a soma dos números pares.
7. Dados $n > 0$ e dois números inteiros positivos i e j diferentes de 0, imprimir em ordem crescente os n primeiros naturais que são múltiplos de i ou de j e ou de ambos.
8. Dados dois números inteiros positivos, determinar o máximo divisor comum entre eles usando o algoritmo de Euclides.

Exemplo:

$$\begin{array}{c|c|c|c|c} & 1 & 1 & 1 & 2 \\ \hline 24 & 15 & 9 & 6 & 3 \\ \hline 9 & 6 & 3 & 0 & \end{array} = \text{mdc}(24,15)$$

9. Dizemos que um número i é congruente módulo m a j se $i \% m = j \% m$.

Exemplo: 35 é congruente módulo 4 a 39, pois $35 \% 4 = 3 = 39 \% 4$.

Dados inteiros positivos n , j e m , imprimir os n primeiros naturais congruentes a j módulo m .

Dúvidas

Dúvidas deste material pode ser enviadas para o “Fórum para assuntos específicos da turma WEB”.