Programação Imperativa
Departamento de Informática
Universidade do Minho

1º Ano LMCC

2004/2005

Ficha Teórico-Prática Nº 3
Estruturas de controlo, strings e conversão de representações

15 de Março de 2005

Objectivos

O objectivo principal desta ficha é familiarizar o aluno com a codificação de algoritmos um pouco mais complexos alguns deles envolvendo manipulação de strings. Outro dos objectivos, é a integração com a disciplina de Arquitectura de Computadores, pelo que no fim da ficha aparece um conjunto de problemas sobre conversão de representações numéricas.

Para atingir esse fim, o aluno irá desenvolver pequenos algoritmos e tentar codificá-los em C.

Nota: os exercícios 2, 3, 4 e 9 são de resolução opcional. Na ficha a submeter ao sistema só são obrigatórios os restantes.

Exercícios

Exercício 1: "Divisores de um número"

Desenvolva o algoritmo, e posteriormente codifique-o em C, de um programa que lê um inteiros e escreve no écran os seus divisores.

Exercício 2: "Maior Divisor Comum"

Desenvolva o algoritmo, e posteriormente codifique-o em C como uma função. A sua função deverá ter dois argumentos inteiros e produzir um resultado inteiro que é o maior divisor comum dos argumentos recebidos. Posteriormente codifique um programa principal (`main()`) que utiliza esta função para cálculo do maior divisor comum de dois números fornecidos pelo utilizador.

Ver `http://mathworld.wolfram.com/EuclideanAlgorithm.html`, Algorítmo de Euclides.

Que pode ser resumido em:

```
mdc(m,n) -->
  m>=n : m%n==0 --> n
```

```
m%n!=0 --> mdc(n,m%n)

 m<n  : mdc(n,m)
```

Exercício 3: ”Menor Múltiplo Comum”

Desenvolva o algoritmo, e posteriormente codifique-o em C como uma função. A sua função deverá ter dois argumentos inteiros e produzir um resultado inteiro que é o menor múltiplo comum dos argumentos recebidos. Posteriormente codifique um programa principal (`main()`) que utiliza esta função para cálculo do menor múltiplo comum de dois números fornecidos pelo utilizador.

Exercício 4: ”É Primo?”

Desenvolva o algoritmo, e posteriormente codifique-o em C como uma função. A sua função deverá ter um argumento inteiros e produzir um resultado do tipo booleano (o docente irá desenvolver uma pequena biblioteca de booleanos: `bool.h`). O resultado será verdadeiro (`TRUE`) se o argumento passado fôr primo e falso (`FALSE`) caso contrário. Posteriormente codifique um programa principal (`main()`) que utiliza esta função para verificar quem é primo numa sequência de inteiros introduzidos pelo utilizador (terminada por 0).

Exercício 5: ”Tudo ao contrário!”

Foste transferido para uma nova escola.

Nessa escola, a professora e matemática gostava muito de pôr os alunos a pensar e tentava que a mais simples tarefa fosse algo em que o raciocínio tivesse que intervir.

Uma das medidas que ela implementou foi a de utilizar os números inteiros invertidos. Por exemplo, o 12 seria 21 e o 1821 seria 1281.

Ora, os alunos locais já estão habituados a lidar com os inteiros daquela maneira e até fazem as operações de adição e subtracção com os inteiros naquela forma. No entanto, para ti, que acabaste de chegar à escola esta tarefa não é tão simples. Assim, resolveste criar um programa em computador que fizesse três coisas:

1. dado um número calcula o seu inverso (o valor inteiro correcto)
2. dados dois números na forma invertida calcula a sua soma e devolve o resultado na mesma forma invertida
3. dados dois números na forma invertida calcula a sua subtracção e devolve o resultado na mesma forma invertida

Sugestão: especifica uma função para cada um dos pontos acima.

Exercício 6: ”Sequência de Fibonacci”

Codifica um programa que gere os primeiros 20 números de Fibonacci. Os números de Fibonacci são definidos recursivamente da seguinte forma:

```
fib(0) = 1
fib(1) = 1
fib(n) = fib(n-1) + fib(n-2)   para n>=2
```

Exercício 7: ”Conversão decimal-binário”

Depois de estudares o algoritmo das divisões sucessivas para a conversão de números decimais em números binários, desenvolvido na aula teórico-prática, codifica-o num programa em C.

Exercício 8: ”Conversão binário-decimal”

Depois de teres resolvido o problema anterior, tenta resolver o problema inverso. Nota que os números binários são representados numa string com os carácteres 0 e 1.

Exercício 9: ”Conversão binário-hexadecimal”

Pensa agora com converter um número binário em hexadecimal. Cria um programa em C para realizar esta tarefa.