

# Programação Imperativa

## LEI + LCC (1º ano)

2ª e 3ª Ficha Prática Complementares

Ano Lectivo de 2006/07

### Objectivos

O objectivo principal desta ficha é familiarizar o aluno com a utilização e manipulação de vectores e matrizes numéricas e com strings em C.

### Exercícios

1. Desenvolva um programa em C que, dada uma sequência de N números, determine a percentagem daqueles que são:
  - a) superiores à média dos elementos que a constituem;
  - b) inferiores ao antepenúltimo elemento da sequência.
2. Desenvolva um programa em C que leia do teclado uma sequência de números inteiros positivos e calcule um conjunto de indicadores sobre a sequência lida. O programa deverá oferecer ao utilizador um menu de operações como o que se mostra a seguir:

Sequências de Inteiros: lista de operações

A - Ler a sequência

B - Escrever a sequência

C - Calcular o máximo da sequência

D - Calcular o mínimo da sequência

E - Determinar a subsequência de números acima da média

F - Determinar a subsequência de números abaixo da média

G - Calcular o mínimo múltiplo comum da sequência

H - Determinar a subsequência dos números que são primos

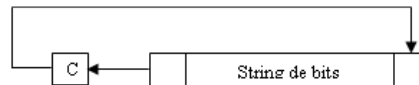
I - Sair do Programa

Opção:

A leitura de números deverá terminar quando for lido o 0 ou quando for atingido o limite da sequência (defina-o). Sugestão: especifique uma função para cada opção do menu inicial.

3. Defina a sua própria versão das seguintes funções da biblioteca das strings em C `string.h`:
  - a) `strcpy(s1,s2)`
  - b) `strcat(s1,s2)`

- c) `strdup(s1)`
  - d) `strndup(s1,n)`
  - e) `atoi(s)`
4. Desenvolva um programa em C que realize as seguintes tarefas:
- a) leia dois números inteiros sem sinal
  - b) transforme esses números em sua respectiva representação em binário
  - c) some as representações binárias dos números
  - d) transforme a soma em binário para decimal e imprima o resultado
5. Desenvolva um programa em C que leia um inteiro e faça a multiplicação e divisão por dois usando a representação binária do número lido. O resultado deve ser apresentado em binário e em decimal.
6. Desenvolva um programa em C que leia um byte (número binário com 8 bits) e implemente a seguinte operação de rotação para a esquerda através do "carry":



Exemplo:

Se a string de bits é: 10101010 e o bit C é 1 o resultado de aplicar a operação deste exercício é: C=1 e a string de bits é 01010101

7. Estenda o programa do exercício anterior (f) de modo a fazer a mesma operação de rotação mas agora para a direita.
8. Um número binário tem paridade par se tem um número par de 1's. Desenvolva um programa em C que determine se um dado byte tem paridade par.
9. Desenvolva a função  $MAPCAR(f,a)$  que tem como parâmetros uma função  $f$  e um array de inteiros  $a$ . A função  $f$  tem como Parâmetro um inteiro e retorna como resultado um inteiro. MAPCAR aplica a função  $f$  a todos os elementos do array  $a$ .  
Por exemplo, Se  $f$  é:

```
int f(int x)
{
    return x+1;
}
```

e  $a$  é:  $a=\{1,2,3,4,5,6\}$

e nós invocamos:  $MAPCAR(f,a)$

O resultado será:  $a=\{2,3,4,5,6,7\}$

10. Desenvolva um programa em C que implemente o tipo de dados conjunto. Este tipo tem as seguintes operações:
- a)  $Member(x,s)$ : retorna como seu resultado "verdadeiro" se o elemento  $x$  está contido no conjunto  $s$  e "falso" em outro caso;

- b)  $Union(dst,s1,s2)$ : esta operação faz a união entre os conjuntos  $s1$  e  $s2$  e o resultado fica o conjunto  $dst$ ;
- c)  $Intersection(dst,s1,s2)$ : esta operação faz a intersecção entre os conjuntos  $s1$  e  $s2$  e o resultado fica o conjunto  $dst$ ;
- d)  $Difference(dst,s1,s2)$ : esta operação faz a diferença entre os conjuntos  $s1$  e  $s2$  e o resultado fica o conjunto  $dst$ .

Para este exercício o aluno deve seleccionar a representação de conjunto que ache mais correcta.

11. Uma procura por frequência de uso é aquela que logo que encontra um elemento na sequência o coloca na primeira posição da sequência para diminuir o tempo de procura desse elemento na próxima vez que seja procurado. Por exemplo, dada sequência (1,2,3,4,5,6,7,8,9,10) se se quer procurar o elemento 10 o tempo de execução é  $N$  onde  $N$  é o comprimento da sequência; mas, se depois dessa procura o elemento 10 é colocado na primeira posição, ou seja depois da procura a sequência fica (10,2,3,4,5,6,7,8,9,1), o tempo de execução para procurar novamente o elemento 10 é 1. Neste contexto pede-se que desenvolva um programa em C que implemente este algoritmo da procura por frequência de uso.
12. Desenvolva em C a função  $operadorBinario(mr,mop1,mop2,f)$  que recebe como parâmetros uma matriz  $mr$  onde ficará o resultado, uma matrix  $mop1$  e outra  $mop2$ , todas quadradas e da mesma dimensão, e uma função binária  $f$ . A função  $operadorBinario$  toma cada elemento das matrizes  $mop1$  e  $mop2$  e aplica a função  $f$ , deixando o resultado na matrix  $mr$ . Por exemplo:

Se a matrix  $mop1$  é:

$$\begin{pmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \end{pmatrix}$$

e a matrix  $mop2$  é:

$$\begin{pmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \\ 70 & 80 & 90 \end{pmatrix}$$

e a função  $f$  implementa a adição de inteiros então  $operadorBinario(mr, mop1,mop2,suma)$  tem o seguinte resultado  $mr$ :

$$\begin{pmatrix} 10 & 21 & 32 \\ 42 & 53 & 64 \\ 75 & 86 & 97 \end{pmatrix}$$