Aulas teórico-práticas de Programação Estruturada

Folha de exercícios

2000/2001

1. *Factoriais*

   Modifique a versão recursiva da função factorial para obter os seguintes "efeitos". Em cada caso a função deve continuar a ser recursiva e não pode haver variáveis globais. Pode, se necessário, aumentar o número de parâmetros da função.

   (a)
   ```
   5         (valor dado)
   fac(5) = 5*fac(4)
   fac(4) = 4*fac(3)
   fac(3) = 3*fac(2)
   fac(2) = 2*fac(1)
   fac(1) = 1*fac(0)
   fac(0) = 1
   120
   ```
   (b)
   ```
   5         (valor dado)
   5*4*3*2*1*1
   120
   ```
   (c)
   ```
   5         (valor dado)
   fac(5) = 5*fac(4)
       fac(4) = 4*fac(3)
           fac(3) = 3*fac(2)
               fac(2) = 2*fac(1)
                   fac(1) = 1*fac(0)
                       fac(0) = 1
                   =1*1=1
               =2*1=2
           =3*2=6
       =4*6=24
   =5*24=120
   120
   ```

2. *Árvores de dois ramos ...* Considere o programa dado nas aulas teóricas onde se ordena um vector pelo seguinte processo: (i) Inserem-se os seus elementos numa árvore binária inicialmente vazia, (ii) Lista-se a árvore em "em-ordem". Ver resumo do método abaixo.

   (a) Copie e teste o programa.

   (b) Altere as ordens da listagem e observe o efeito (pré-ordem e pós-ordem).

   (c) Altere o tipo dos elementos armazenados: passam a ser pares de "strings" representando pares do tipo (nome,telefone), sendo os nomes constituídos por uma só palavra.

   (d) Altere o programa da alínea anterior para funcionar do seguinte modo: (i) Os elementos a inserir são lidos de um ficheiro cujo nome é dado na linha de comando, (ii) O utilizador fornece repetidamente um nome e o computador imprime o respectivo telefone ou "???" se o nome não existe.

Usando uma árvore binária de pesquisa para ordenar um vector

Noção de árvore binária (definição indutiva): ou é nula ou é da forma:

(árvore binária, raiz, árvore binária)

Modos de visita:

- Pré-ordem: Raiz, sub-árvore esquerda, sub-árvore direita (recursivamente).
- Em-ordem: Sub-árvore esquerda, raiz, sub-árvore direita (recursivamente).
- Pós-ordem: Sub-árvore esquerda, sub-árvore direita, raiz (recursivamente).

Um modo de ordenar vectores:

(a) Inserir os elementos a ordenar numa árvore binária.
(b) Listar a árvore em-ordem.

```
typedef struct no{
  int valor;
  struct no *esq;
  struct no *dir;
} NO, *ARVORE;

/* insert: Arv x N -> Arv        */
ARVORE insert(int,ARVORE);
void em_ordem(ARVORE);
/* Exemplo:                      */
int v[]={55,12,88,3,21,4};

main()
{
  ARVORE a=0;
  int i;
  for(i=0;i<6;i++)
    a=insert(v[i],a);
  em_ordem(a);
}
/*----------------------------------------------------*/
ARVORE insert(int x,ARVORE a)
{
  if(a==0){
    ARVORE nova;
    nova=(ARVORE)malloc(sizeof(NO));
    nova->valor = x;
    nova->esq   = 0;
    nova->dir   = 0;
    return(nova);
  }
  if(x <= a->valor){
    a->esq = insert(x,a->esq);
    return(a);
  }
  a->dir = insert(x,a->dir);
  return(a);
}

/*------------------------------------------------------*/
void em_ordem(ARVORE a)
{
  if(a==0)
    return;
  em_ordem(a->esq);
  printf(" %d ",a->valor);
  em_ordem(a->dir);
}
```