

Transcrição de aula

Disciplina	Programação Imperativa - 1º ano - LEI	
Secretário	Número: 61085	Nome: Pedro Casimiro
Data: 2011-05-02	Nº Página 4	

J. Ramalho

SUMÁRIO

Algoritmos sobre listas ligadas.
O caso dos números "gigantes".

```

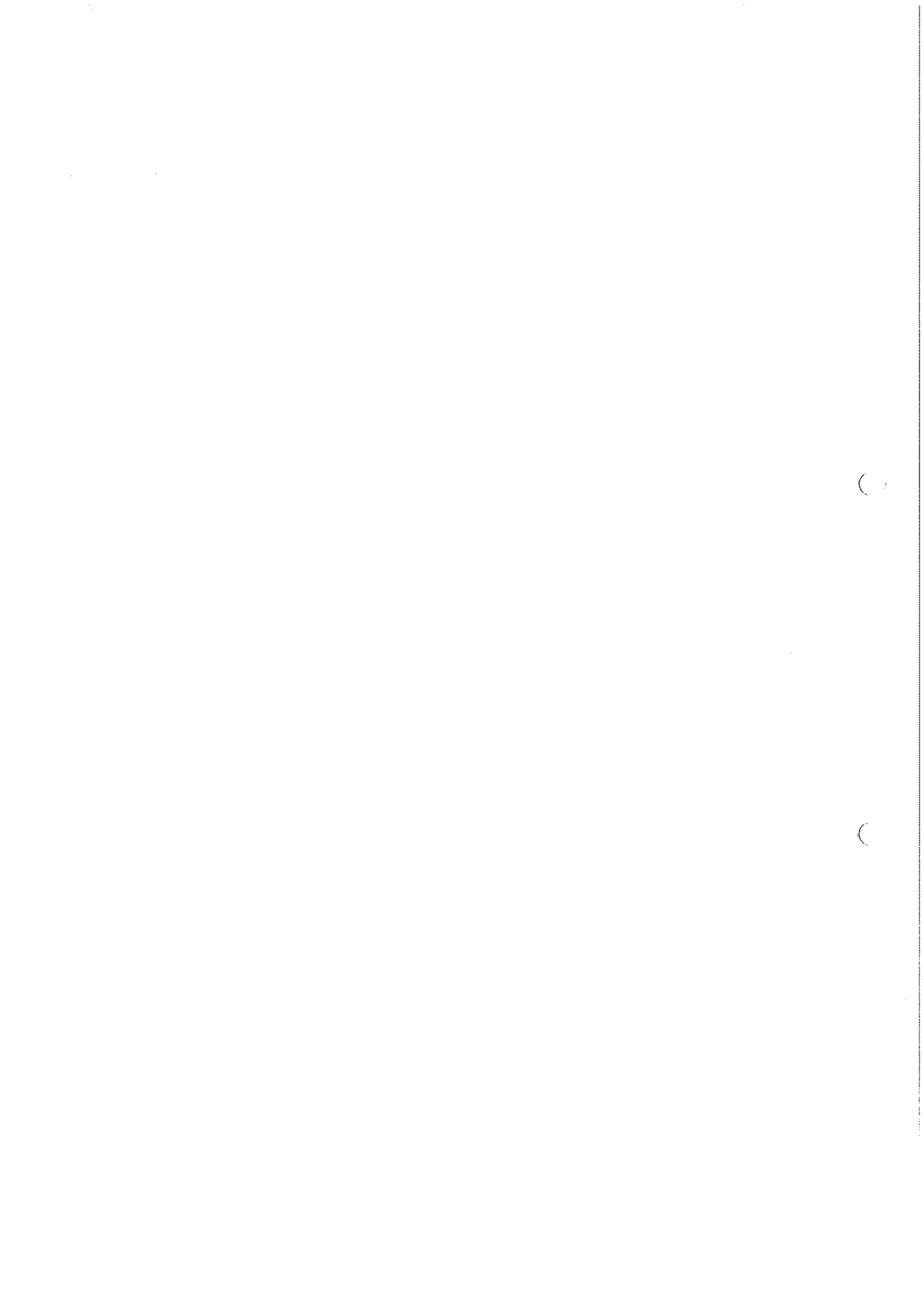
Números Gigantes
typedef struct sInt
{
    int valor;
    struct sInt *seg;
} *Numero, Digito;

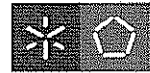
Numero SumBigInt (Numero m1, Numero m2);
Numero SumBigInt (Numero m1, Numero m2,
                  int carry);
Numero Double (Numero m);

Numero SumBigInt (Numero m1, Numero m2)
{
    return SumBigInt (m1, m2, 0);
}

Numero SumBigInt (Numero m1, Numero m2, int carry)
{
    Numero ns;
    if (!m1)
        ns = Double (m2);
    else
        if (!m2)
            ns = Double (m1);
}

```





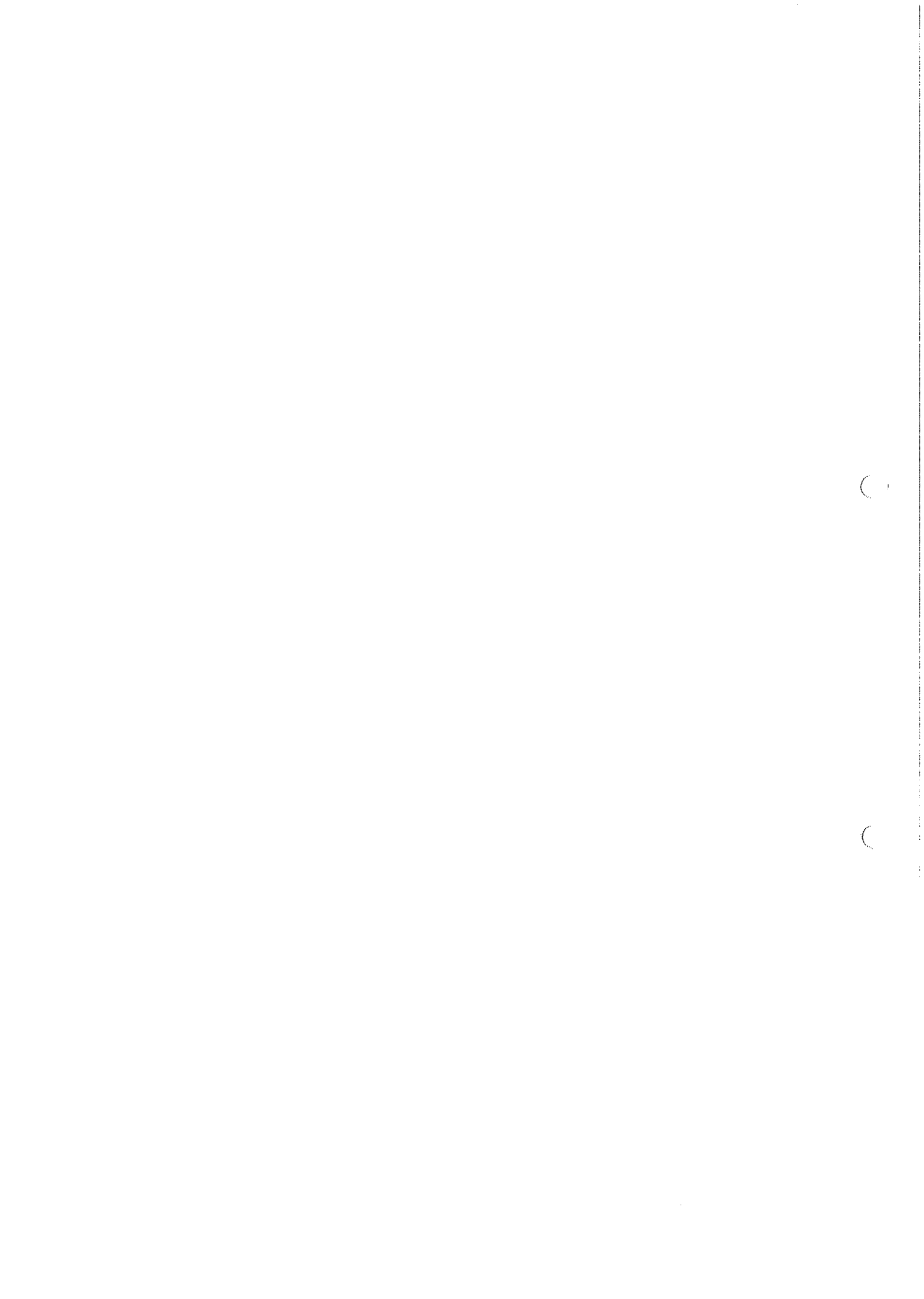
Transcrição (folha de continuação)

```
elre
{
  res = (Numero) alloc (sizeof (Digito));
  soma = m1 ->valor + m2 ->valor + carry;
  res ->valor = soma % 10;
  carry = soma / 10;
  res ->seg = SumBigNum (m1 ->seg, m2 ->seg, carry);
}
return res;
}

Numero Double (Numero m)
{
  Numero aux = NULL;
  if (m)
  {
    aux = (Numero) alloc (sizeof (Digito));
    aux ->valor = m ->valor;
    aux ->seg = Double (m ->seg);
  }
  return aux;
}

typedef struct stInt
{
  int valor;
  struct stInt *seg;
} * lista, ** lista2;

int GetElem (lista l, int index);
lista Inserir (lista l, int index);
lista Append (lista l1, lista l2);
```





Transcrição (folha de continuação)

```
int GetCellh (lista l, int index)
{
  if (!index)
    return l->valor;
  else
    return GetCellh (l->prox, index-1);
}

lista InverteCellh (lista l, int v, int index)
{
  if (!index)
    l = Push (l, v);
  else
    l->prox = InverteCellh (l->prox, v, index-1);
  return l;
}

lista Append (lista l1, lista l2)
{
  if (!l1)
    return l2;
  else
  {
    l1->prox = Append (l1->prox, l2);
    return l1;
  }
}

(continua no verso)
```

```
int void inc2 (int m)
{
    m += 2;
}
void inc3 (int *m)
{
    *m += 3;
}
```

```
int main ()
{
    int a = 0;
    inc3 (&a);
    ...
    inc2 (a);
}
```