



# Transcrição de aula

TP<sub>2</sub>  
(40 alunos)

Disciplina	Programação Imperativa - 1º ano - LEI	
Secretário	Número: A56859	Nome: NUNO JORGE M. SANTOS
Data: 2011-05-01	Nº Página	

## SUMÁRIO

$a^b$  Problema: Escrever uma função que calcule o exponencial

$a=4$   
 $b=3$   
 $4 * 4 * 4$

```
int pot (int a, int b)
{
  int res = 1;
  while (b > 0)
  {
    res = a * res;
    b --;
  }
  return res;
}
```

```
int expres (int a, int b)
{
  if (b == 0) return 1;
  else return expres (a, b-1);
}
```



### Transcrição (folha de continuação)

mdc (máximo divisor comum)

16   2 8   2 4   2 2   2 1	12   2 6   2 3   3 1	16 12 -12 (4)	48 18 30 18 18 12 12 (6)	% 48 18 18 12 12 (6)
--	-------------------------------	------------------	-----------------------------------	-------------------------------

```
int mdc (int a, int b)
{
  int aux;
  while (a % b != 0)
  {
    aux = a;
    a = b;
    b = (aux % b);
  }
  return b;
}
```

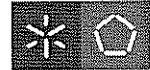
```
int mdc (int a, int b)
{
  if (a % b == 0)
    return b;
  else mdc (b, (a % b));
}
```

---

Um função que pegue um numerador e um denominador, e os reduza (usando o mdc)

```
int main ()
{
  int a, b, x;
  printf ("Introduza o numerador:");
  scanf ("%d", &a);
  printf ("Introduza o denominador:");
  scanf ("%d", &b);
  x = mdc (a, b);
  printf ("%d / %d", a/x, b/x);
  return 1;
}
```

```
if (a > b)
  x = mdc (a, b);
else
  x = mdc (b, a);
```



### Transcrição (folha de continuação)

```
int primo (int n)
{
  int teste = n/2, out = 0;
  while ((teste >= 2) && (out != 1))
  {
    if ((n % teste) == 0)
      out = 1;
    teste--;
  }
  return !out;
}
```

0 - não é primo  
1 - é primo  
divisores possíveis:  
[2, n/2]

```
int vogal (char c)
{
  int res = 0;
  switch (c)
  {
    case 'a':
    case 'e':
    case 'i':
    case 'o':
    case 'u': res = 1;
  }
  return res;
}
```

```
0 - não é vogal  
1 - é vogal  
Switch (<var>)
{
  case <valor>: <ação>
               <ação>
               break;
  case <valor>: ...
  default: <ação>
           <ação>
           ...
}
```

```
if (c == 'a' || c == 'e' || ...)
  res = 1;
return res;
```

iguais



### Transcrição (folha de continuação)

```
int maxSeq (int A[], int nelems)
{
    int i = 1, b = A[0];
    while (i < nelems)
    {
        if (A[i] >= b) { b = A[i]; }
        i++;
    }
    return b;
}
```

```
int maxSeqrec (int A[], int nelems)
{
    return maxSeqaux (A+1, nelems-1, A[0]);
}
int maxSeqaux (int A[], int n, int max)
{

```

TPC

```
}
```