

# DIM0108.0 - 2011.2 - Turma 1 - Prova 03

David Déharbe - UFRN/CCET/DIMAp

6 de Dezembro de 2011

Considere o seguinte código fonte:

```
#include <stdio.h>
int mist(char * s, char c)
{
    int res = 0;
    while (s[res] != '\0' && s[res] != c) {
        res += 1;
    }
    if (s[res] == c)
        return res;
    return -1;
}

int main (void)
{
    char w[10] = "oi";
    printf("%s: %i %i %i\n", w, mist(w,'o'), mist(w,'i'), mist(w, 'u'));
    return 0;
}
```

**Exercício 1** *Qual o resultado da execução deste programa?* 1,0

**Exercício 2** *Descreva o papel da sub-rotina mist.* 1,0

**Exercício 3** *Defina uma sub-rotina que testa se um valor do tipo char representa uma letra (minúscula ou maiúscula).* 1,0

**Exercício 4** *A sub-rotina pré-definida toupper tem como parâmetro um caracter e retorna um caracter. Se o argumento for uma letra minúscula, retorna a letra maiúscula correspondente. Caso contrário, retorna o próprio caracter. Você pode usar esta sub-rotina para resolver este exercício.*

*Defina uma sub-rotina strupp que tem como parâmetros str o endereço do primeiro caracter de um texto e retorna um novo texto correspondente à uma cópia do texto apontado por str onde todas as letras minúsculas foram substituídas pela letra maiúscula correspondente.* 1,5

*Por exemplo, strupp("Ola!") deve retornar o endereço de um bloco de memória armazenando o texto "OLA!".*

Considere o seguinte programa:

```
#include <stdio.h>

typedef struct TSpoin {
    double x;
    double y;
} Tpoint;

int main(void)
{
    Tpoint p = {-0.25, 1.5};
    printf("[%f %f]\n", p.x, p.y);
    return 0;
}
```

**Exercício 5** Defina uma sub-rotina `point_printf` tal que o programa seguinte seja equivalente ao programa anterior. 1.0

```
#include <stdio.h>

typedef struct TSpoin {
    double x;
    double y;
} Tpoint;

int main(void)
{
    Tpoint p = {-0.25, 1.5};
    point_printf(p);
    return 0;
}
```

**Exercício 6** Defina uma sub-rotina `point_dist` que calcula a distância entre dois pontos, representados com o tipo `Tpoint`. 1.0

**Exercício 7** Defina um tipo `Trect` para retângulos cujos lados são paralelos aos eixos do sistema de coordenadas cartesianas. O retângulo deve ser representado pelos cantos inferior esquerdo e superior direito, representados com o tipo `Tpoint`.

**Exercício 8** Defina uma sub-rotina que retorna a área de um retângulo representado com o tipo `Trect`.

**Exercício 9** Defina uma sub-rotina que testa se um ponto, representado com o tipo `Tpoint`, é interno a um retângulo, representado com o tipo `Trect`.