

TP4 - Trabalho prático de CI1001 e CI067  
Trabalho prático 4 - Introdução à linguagem C  
Departamento de Informática/UFPR  
Albini, Bona, Castilho e Grégio

Esse trabalho prático é dividido em 3 subtarefas e seu objetivo exercitar e compreender o conceito de ponteiros. Entregue um único arquivo de nome `tp4.tar.gz` no sistema moodle contendo os arquivos de respostas conforme os nomes abaixo. Seus programas devem compilar sem erros com as flags `-std=C90 -Wall`.

As tarefas deste trabalho foram parcialmente baseada no exercício encontrado em:

<http://www.csc.villanova.edu/~mdamian/Past/csc2400fa13/assign/plab.pdf>.

Nesse documento vocês encontram algumas dicas úteis e outros exercícios complementares.

## 1 `tp4.1.c`

Escreva um programa em C que declare e inicialize (com qualquer valor) uma variável do tipo `double`, uma do tipo `int` e uma do tipo `char`. Em seguida, declare um ponteiro para cada uma das três variáveis. Lembre-se de inicializar cada ponteiro para a variável correspondente.

Seu programa deve então imprimir o endereço, o valor armazenado e o tamanho da memória (em bytes) de cada uma das seis variáveis. Considere que:

- A impressão dos endereços deve ser feita usando a notação hexadecimal, use a string de formatação `%p` no `printf`;
- O comando `sizeof` permite saber o tamanho ocupado por um determinado tipo;

A saída esperada para o programa é (imprima os nomes das variáveis que você declarou e as demais informações pedidas):

```
0 endereço de char ___ é 0x_____  
0 endereço de int ___ é 0x_____  
0 endereço de double ___ é 0x_____  
0 endereço de char* ___ é 0x_____  
0 endereço de int* ___ é 0x_____  
0 endereço de double* ___ é 0x_____
```

```

O valor de char ___ é _____
O valor de int ___ é _____
O valor de double ___ é _____
O valor de char* ___ é 0x_____
O valor de int* ___ é 0x_____
O valor de double* ___ é 0x_____

O tamanho de char é _____ bytes
O tamanho de int é _____ bytes
O tamanho de double é _____ bytes
O tamanho de char * é _____ bytes
O tamanho de int* é _____ bytes
O tamanho de double* é _____ bytes

```

## 2 tp4.2.c

No programa abaixo a função `troca_nums` funciona corretamente enquanto a função `troca_ponteiros` não. Corrija o programa e faça comentários explicando suas alterações.

```

#include <stdio.h>

void troca_nums(int *x, int *y)
{
    int tmp;
    tmp = *x;
    *x = *y;
    *y = tmp;
}

void troca_ponteiros(char *x, char *y) {
    char *tmp;
    tmp = x;
    x = y;
    y = tmp;
}

```

```

int main()
{
    int a,b;
    char *s1,*s2;
    a=3;
    b=4;
    troca_nums(&a,&b);
    printf("a eh %d\n", a);
    printf("b eh %d\n", b);
    s1 = "Imprimir isso primeiro";
    s2 = "Imprimir isso depois";
    troca_ponteiros(s1,s2);
    printf("s1 eh %s\n", s1);
    printf("s2 eh %s\n", s2);
    return 0;
}

```

### 3 tp4.3

Complete o programa abaixo conforme os comentários.

```

#define TAM 10

int main()
{
    int vetor[TAM];
    char abc[]={'a','b', 'c', 'd'};
    void *p;
    int x;

    /* p aponta para vetor */
    p = (void*)vetor;

    /* escrevendo caracteres no vetor usando p */
    for (x=0;x< TAM*4;x++)
    {
        *(char*)p = abc[x%4];
        p=(char*)p+1;
    }
}

```

```

/* apontando novamente para o inicio do vetor */
p = (void*)vetor;

/* imprimindo p como um vetor de inteiros */
for (x=0;x<TAM;x++)
{
    printf("(%p) vetor[%d] = %d \n", p, x, *(int*)p);
    p=(int*)p+1;
}
printf("\n");

/* imprimindo p como um vetor de caracteres */

/* seu codigo */

return 0;
}

```