

Programação I

Apontadores (ficha 14)

1. Escreva uma função que recebe dois apontadores para inteiros (passagem de parâmetros por referência) e devolve o endereço do maior valor. Escreva um programa para testar a função. Considere o protótipo `int *vmaior(int *v1, int *v2)`.
2. Escreva a função `void ordena(int *a, int *b, int *c)` que recebe 3 valores e ordena-os, colocando o menor em `a` e o maior em `c`.
3. O programa seguinte tem como objetivo imprimir os valores de `var1` e `var2` e os seus endereços (guardados em `ptr1` e `ptr2`).

```
#include <stdio.h>

int main(){
    int var1=5;
    char var2='a';
    int *ptr1= &var1;
    char *ptr2;
    *ptr2='b';
    printf("var1 tem o endereço %p e o valor %d\n", ptr1, var1 );
    printf("var2 tem o endereço %p e o valor %c\n", ptr2, *ptr2 );
}
```

Siga os seguintes passos e interprete, para cada caso, os valores apresentados pelo programa:

- (a) Se tentar compilar e executar este programa ocorrerá o erro `Segmentation fault`. Corrija o programa para que isso não aconteça.
 - (b) Altere o programa para que este imprima também o tamanho dos tipos de variáveis `char`, `char *`, `int` e `int *`.
 - (c) Altere o programa para que este imprima ainda os valores de `prt1+1` e `ptr2+1` e compare com os valores de `ptr1` e `ptr2`.
4. Considere o programa seguinte e complete as tabelas.

```
#include <stdio.h>

int main(){
    int i, j, *p1, *p2, **pp1, **pp2;
    i=4;
    j=5;
    p1 = &i;
    p2 = &j;
    pp1 = &p2;
    pp2 = &p1;
}
```

variavel	i	j	p1	p2	pp1	pp2
conteudo	4	5				
endereço	1000	1007	1030	1053	1071	1079

expressão	i	*p2	&i	&p2	*pp1	*pp2	&(*p1)	j	*p1	*(&p1)
conteúdo										