

. **ATIVIDADE DE FIXAÇÃO 2**

Informações importantes:

- 1. Os exercícios que não pedirem a implementação de código favor descrever a resposta e anexar como comentário no código fonte.
- 2. Em todos os exercícios, caso entenda necessário ou que fique melhor, vocês podem fazer mais de um predicado para resolver o problema.
- 3. Para todos os exercícios vocês devem fazer um comentário no código explicando como executar o código citando exemplos.
- 4. Para todas as funções recursivas devem ser descritas o seu funcionamento.

1)

```
void altera1(int **p, int *a)
{
    **p = *a;
    *a = *a +50;
}

void altera2(int **p, int *b)
{
    *p = b;
    *b = *b +300;
}

int main()
{
    int x,y, *px, *py;

    x = 500;
    y = x + 200;

    px = &x;
    py = &y;

    printf("x = %d, End. x = %p, px = %p, y = %d \n",x,&x, px,y);
    altera1(&px, &y);
    printf("x = %d, End. x = %p, px = %p, y = %d \n",x,&x, px,y);
    getchar();

    printf("y = %d, End. y = %p, py = %p, x = %d \n",y,&y, py,x);
    altera2(&py, &x);
    printf("y = %d, End. y = %p, py = %p, x = %d \n",y,&y, py,x);
    getchar();

    return(0);
}
```

Dado o código acima, responda as seguintes questões:

(a) Qual a diferença entre px e x?

R: px é um ponteiro e x é uma variável comum do tipo inteiro.

(b) Qual a diferença entre px e py?

R: px é um ponteiro que aponta para endereço de memória da variável x e py é um ponteiro que aponta para o endereço de memória da variável Y

(c) Quais são os valores impressos pelo primeiro printf?

R: x = 500, End. x = 0x7ffc784cfde0, px = 0x7ffc784cfde0, y = 700

(d) O que muda do primeiro printf para o segundo?

R: muda os valores de x que antes era 500 e no segundo printf x=700 e o valor de y que antes era 700 e no segundo printf y=750 .

(e) Quais os valores impressos pelo terceiro print?

R: y = 750, End. y = 0x7ffc784cfde4, py = 0x7ffc784cfde4, x = 700

(f) O que muda do terceiro para o quarto print?

R: Muda o endereço de memória que py está apontando e valor de x que antes era 700 e no quarto printf x=1000 .

(g) Explique a diferença entre o altera1 e o altera2.

R: A principal diferença é que na função altera 1 o ponteiro **p = *a recebe o valor que está contido no ponteiro *a e o ponteiro *a recebe o valor da soma *a+50, já na função altera2 o ponteiro *p recebe o endereço de memória da variável b e *b recebe o valor de *b+300.

- 2) Refaça o código acima agora eliminando as variáveis x e y. Usando o comando malloc para reservar o endereço para os ponteiros px e py. Lembre-se que as funções altera1 e altera2 terão que ser modificadas pois agora só devem ter um único parâmetro. Os prints devem imprimir as mesmas informações do exercício anterior.
- 3) Faça um programa em C que leia dados de uma Pessoa (CPF e Nome), então armazene os dados em um ponteiro. Imprima o endereço de memória alocado e os dados da Pessoa. Depois faça uma função que permita alterar o nome de uma pessoa. Depois de alterar o nome da pessoa, imprima novamente o endereço de memória alocado e os dados da pessoa. Ao final, libere o espaço de memória alocado. O código deve permitir o usuário executar o código quantas vezes desejar.

Obs1.: Para liberar espaço de memória pesquise sobre o comando free.

Obs2.: Permitir o usuário executar quantas vezes desejar o código tem como objetivo vocês observarem que cada vez um endereço de memória diferente será alocado.