

Introdução aos ponteiros

1. O exemplo que segue apresenta um exemplo clássico de como usar os ponteiros, para manipular os valores de outras variáveis:

```
#include <stdio.h>
int main() {
    int preco = 52;
    int *p;

    printf("preco=: %d\n", preco);

    p = &preco;
    printf("preco=: %d\n", preco);

    *p=65;
    printf("preco=: %d\n", preco);

    printf("p=: %p\n", p);
    printf("&p=: %p\n", &p);
    return 0;
}
```

2. O código abaixo apresenta o conceito básico dos ponteiros:

```
#include <stdio.h>
int main() {
    int i = 100;
    int *p = &i; // int *p; p=&a;
    *p = 200;

    printf("i=: %d\n", i);
    printf("*p=: %d\n", *p);
    printf("endereço de i=: %p\n", &i);
    printf("endereço de p=: %p\n", &p);

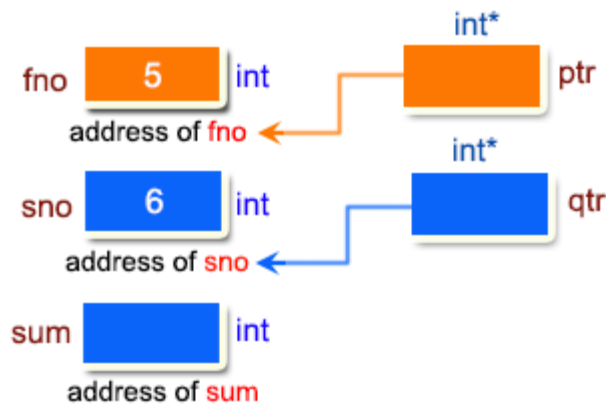
    return 0;
}
```

Pretende-se que o programa seja modificado com as seguintes alterações:

- Introduzir o ponteiro para inteiros: **ptr*

- Antes da instrução *return 0* assignar o endereço de *i* a este novo ponteiro
- A seguir à instrução anterior alterar o valor do endereço que se encontra em *ptr* para 500 e escrever o valor da variável *i*

3. Desenvolver um programa em C que efetua a soma de dois valores(lidos do teclado) através de ponteiros:



4. Implementar um programa de acordo que as seguintes especificações:
- Definir a variável *ch*, do tipo *char* e assignar o valor *T*
 - Escrever no terminal o endereço e valor desta variável
 - Definir uma variável ponteiro *ptr*, que armazena o endereço da variável anterior
 - Escrever o valor do endereço que se encontra em *ptr*
 - Assignar o valor *G* a *prt* e escrever o valor de *ch*

5. O valor NULL é um literal que representa uma referência nula, e que não se refere a nenhum objeto, nenhuma área de memória.

```
#include <stdio.h>
int main() {
    int a = 42;
    int b = 7;
    int c = 999;
    int *t = &a;
    int *u = NULL;

    printf("%d %d\n", a, *t);

    c = b;
    u = t;
    printf("%d %d\n", c, *u);

    a = 8;
    b = 8;
    printf("%d %d %d %d\n", b, c, *t, *u);

    *t = 123;
    printf("%d %d %d %d\n", b, c, *t, *u);

    return 0;
}
```