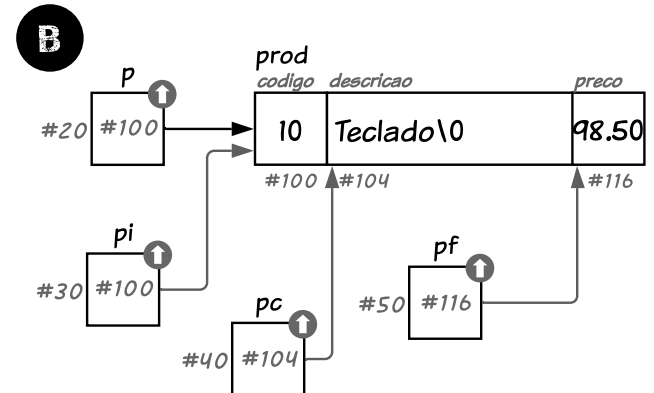
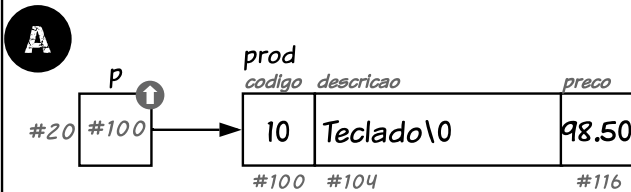


Struct 01

```
typedef struct produto{
    int codigo;
    char descricao[12];
    float preco
} Produto;
```

```
Produto prod = {1, "Teclado", 98.50};
Produto* p = &prod;
```



Determine os valores com base na ilustração A

p : #100	prod.preco: 98.50
&p : #20	&prod.preco: #116
&prod : #100	p->preco: 98.50
prod.codigo: 10	&p->preco: #116
&prod.codigo: #100	
p->codigo: 10	
&p->codigo: #100	
prod.descricao: Teclado	
&prod.descricao: #104	
p->descricao: Teclado	
&p->descricao: #104	

Escreva um trecho de código que reproduza a ilustração B.

```
typedef struct produto{
    int codigo;
    char descricao[12];
    float preco
} Produto;

Produto prod = {1, "Teclado", 98.50};
Produto* p = &prod;

int* pi = &prod.codigo;
// int* pi = &p->codigo;

char* pc = &prod.descricao;
// char* pc = &p->descricao;

float* pf = &prod.preco;
// float* pf = &p->preco;
```

Considerando a ilustração B, faça o que se pede

* Altere o valor do CÓDIGO do produto. Escreva uma linha de código para cada forma possível.

```
prod.codigo = 200;
p->codigo = 200;
*pi = 200;
pi[0] = 200;
```

* Altere o valor da DESCRIÇÃO do produto. Escreva uma linha de código para cada forma possível.

```
strcpy("mouse", prod.descricao);
strcpy("mouse", p->descricao);
strcpy("mouse", pc);
```

* Altere o valor do PREÇO do produto. Escreva uma linha de código para cada forma possível.

```
prod.preco = 105.60;
p->preco = 105.60;
*pf = 105.60;
pf[0] = 105.60;
```