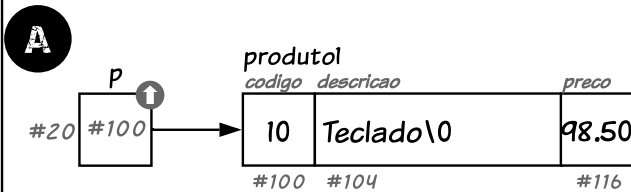


Struct 01

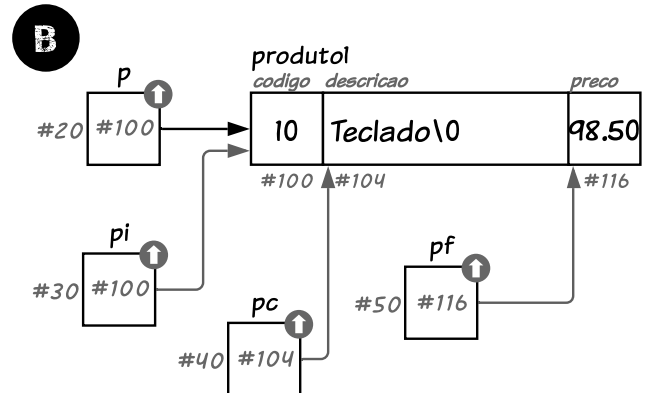
```
typedef struct produto{
    int codigo;
    char descricao[12];
    float preco
} Produto;

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



Determine os valores com base na ilustração A

p :	<u>#100</u>	prod.preco:	<u>98.50</u>
&p :	<u>#20</u>	&prod.preco:	<u>#116</u>
&prod :	<u>#100</u>	p->preco:	<u>98.50</u>
prod.codigo:	<u>10</u>	&p->preco:	<u>#116</u>
&prod.codigo:	<u>#100</u>		
p->codigo:	<u>10</u>		
&p->codigo:	<u>#100</u>		
prod.descricao:	<u>Teclado\0</u>		
&prod.descricao:	<u>#104</u>		
p->descricao:	<u>Teclado\0</u>		
&p->descricao:	<u>#104</u>		



Escreva um trecho de código que reproduza a ilustração B.

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

Produto prod = {10, "Teclado", 98.50};
Produto* p = (Produto*)&prod;
Produto* pi = (Produto*)&prod;
Produto* pc = (Produto*)&prod.descricao;
Produto* pf = (Produto*)&prod.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.
- * Altere o valor da DESCRIÇÃO do produto. Escreva uma linha de código para cada forma possível.
- * Altere o valor do PREÇO do produto. Escreva uma linha de código para cada forma possível.

```
prod.codigo = 11;
(*p).codigo = 11;
p->codigo = 11;
(*pi).codigo = 11;
pi->codigo = 11;
strcpy(prod.descricao, "Teclado G");
strcpy((*pc).descricao, "Teclado G");
strcpy(pc->descricao, "Teclado G");
prod.preco = 200.99;
(*pf).preco = 200.99;
pf->preco = 200.99;
```