

## Struct 02

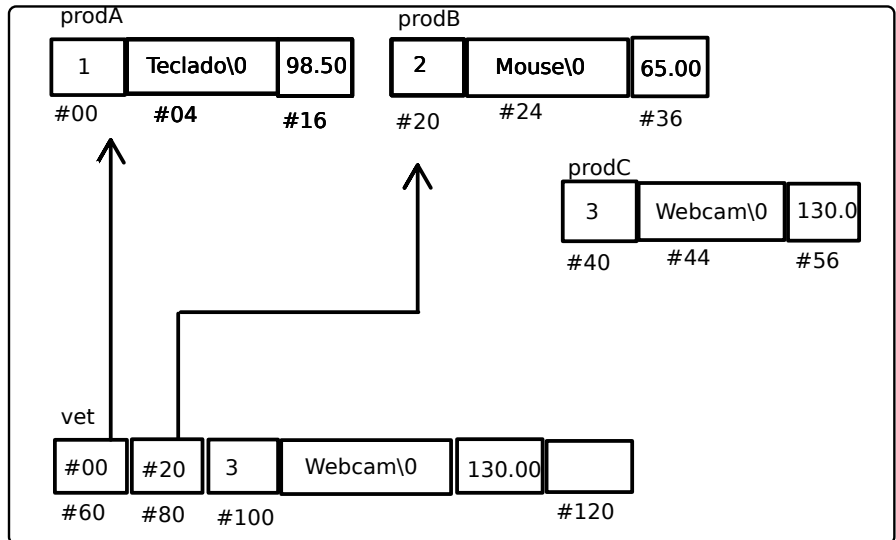
Represente graficamente a organização dos dados na memória dos trechos de código abaixo. Utilize a mesma representação gráfica utilizada durante as aulas

A

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

Produto prodA = {1, "Teclado", 98.50};
Produto prodB = {2, "Mouse", 65.00};
Produto prodC = {3, "Webcam", 130.00};

Produto vet[4];
vet[0] = prodA;
vet[1] = prodB;
vet[2].codigo = prodC.codigo;
strcpy(vet[2].descricao, prodC.descricao);
vet[2].preco = prodC.preco;
```

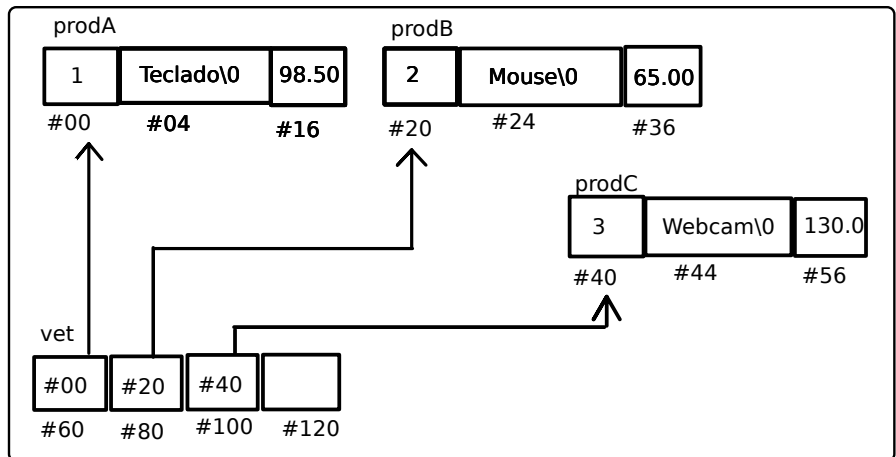


B

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

Produto prodA = {1, "Teclado", 98.50};
Produto prodB = {2, "Mouse", 65.00};
Produto prodC = {3, "Webcam", 130.00};

Produto* vet[4];
vet[0] = &prodA;
vet[1] = &prodB;
vet[2] = &prodB;
```



A partir do código e da ilustração B, faça o que se pede.

- \* Altere o código do produto C para 100. A alteração deve ocorrer por meio do vetor.
- \* Altere a descrição do produto B para "Mouse sem fio". A alteração deve ocorrer por meio do vetor.
- \* Crie uma nova variável de produto, preencha todos os seus dados e atribua para a posição 3 do vetor.

```
*(vet[2]).codigo = 100;
strcpy(*(vet[1]).descricao, "Mouse gamer");
Produto prodD = {4, "Headphone", 300.00};
vet[3] = &prodD;
```