



Exercício:

Vamos utilizar o mesmo tipo Vector do exercício anterior para implementar um vetor de ponteiros genéricos. Assim, seu vetor poderá receber dados de todos os tipos.

Adicione a seguinte enum ao seu vectoritem.h:

```
typedef enum{
    INT = 0,
    FLOAT = 1,
    DOUBLE = 2,
    CHAR = 3
} ItemType;
```

Após isso, modifique suas funções dos tipos VectorItem e Vector para receberem diferentes tipos de dados, executando suas tarefas normalmente.

Para testar, utilize a main abaixo:

```
#include <stdio.h>
#include "vector.h"

int main(){
    Vector *v = vector_create();

    int qntdNumeros;

    scanf("%d", &qntdNumeros);

    for(int i = 0; i < qntdNumeros; i++){
        vector_add(v, &i, 0);
        vector_print(v);
    }

    char c = 'c';
    double d = 3.0;
    float f = 4.5;

    vector_add(v, &c, CHAR);
    vector_add(v, &d, DOUBLE);
    vector_add(v, &f, FLOAT);
}
```

```
vector_print(v);  
vector_remove(v, qntdNumeros + 2);  
vector_print(v);  
  
vector_destroy(v);  
}
```

Entradas	Saídas
3	[0] [1, 1] [2, 2, 2] [0, 0, 0, c, 3.000000, 4.500000] [0, 0, 0, c, 3.000000]