CPSC326 | Homework 7: O' Camel | 4/25/2025 | Arjuna Herbst

Tests for each function:

```
# my_min [-1;2;3;5;0];;
- : int = -1
# my_min [];;
Exception: Failure "Empty List".
```

I used these test cases to ensure my function could handle negative ints and properly identify them as the minimum value. I also input an empty list to make sure my function handled it properly.

```
# my_reverse [1; 2; 3];;
- : int list = [3; 2; 1]
# my_reverse [];;
- : 'a list = []
```

Simple int list, empty list to ensure my function would let it pass through properly.

```
# my_take 0 [1; 2; 3];;
- : int list = []
# my_take 2 [1; 2; 3];;
- : int list = [1; 2]
# my_take 4 [1; 2; 3];;
- : int list = [1; 2; 3]
```

0 as index to make sure function would return empty list, 4 since it's bigger than the size of the list to ensure my function would return the entire list and not throw an error.

```
# my_drop 0 [1; 2; 3];;
- : int list = [1; 2; 3]
# my_take 3 [1; 2; 3];;
- : int list = [1; 2; 3]
```

Same as above for my_take.

```
# my_set 2 5 [1; 2; 3; 2];;
- : int list = [1; 2; 5; 2]
# my_set 3 9 [1; 2; 3];;
Exception: Failure "Invalid Index".
```

Simple int list to make sure function works properly, then passing an index outside the list to ensure function throws exception.

```
# my_init [1; 2; 3];;
- : int list = [1; 2]
# my_init [];;
Exception: Failure "Empty List".
```

Simple int list to make sure function works properly, empty list to make sure function throws exception.

```
# kv_build [1; 2; 3; 2] [true; false; true; true];;
- : (int * bool) list = [(1, true); (2, false); (3, true); (2, true)]
# kv_build [] [1; 2];;
- : ('a * int) list = []
```

Lists with different types to make sure function can handle them, empty list to ensure function returns empty list.

```
# kv_key 'a' [('a', 1); ('b', 2)];;
- : bool = true
# kv_key 'c' [('a', 1); ('b', 2)];;
- : bool = false
```

Don’t really have much to say about these.

```
# kv_remove 'a' [('a', 1); ('b', 2); ('a', 3)];;
- : (char * int) list = [('b', 2)]
# kv_remove 1 [];;
- : (int * 'a) list = []
```

First input tests my function works properly, empty list to make sure function returns empty list.

```
# kv_collect [('a',1); ('b',2); ('a',3); ('c',4)];;
- : (char * int list) list = [('a', [1; 3]); ('b', [2]); ('c', [4])]
# kv_collect [];;
- : ('a * 'b list) list = []
```

Same as above for kv_remove.