**Q1**.

The Result of A (**let**) is:

|  |
| --- |
| Index 0  Index 1  Index 2  Index 3  Index 4  Index 5  Index 6  Index 7  Index 8  Index 9 |

The Result of B (**var**) is:

|  |
| --- |
| Index 10 |

Firstly, we have to define some differences between let and var in general:

* When **Let** and **var** are in a function scope, but if we define them in an (**for loop**) , then **var** variable will be visible to the whole function while **let** will be visible only in the **for**() parentheses.
* Assuming strict mode, **var** will let you re-declare the same variable in the same scope. On the other hand, **let** will not.

So, the Explanation will be as:

With **var** you have a function scope, and only one shared binding for all of your loop iterations. So the ***i*** in every setTimeout callback means the same variable that finally is equal to 10 after the loop iteration ends and get the last change to ***i*** in key++.

While with let you have a **block** scope and when used in the for loop you get a new binding for each iteration So the i in every setTimeout callback means a different variable, each of which has a different value: the first one is 0, the next one is 1 etc.