1. Не очень разобрался со всеми требуемыми типами, поэтому использовал такой json массив:

[{"type":"text","label":"Test","req":1},

{"type":"textarea","label":"123456789","req":0},

{"type":"select","label":"Example","req":0,"choices":[{"label":"1","sel":0},

{"label":"2","sel":0},

{"label":"3","sel":0}]}]

Получившуюся форму можете найти в файле index.html

2. **Первая функция** реализовывает бинарный поиск в массиве.

Эту функцию можно оптимизировать, добавив проверку середины отрезка на искомое значение.

static int Func1(KeyValuePair<int, string>[] a, int low, int high, int key)

{

  int middle = low + ((high - low)/2);

  if (low == high)

    return low;

int middleValue = a[middle].Key;

if (middleValue == key)

return middle;

  if (key > a[middle].Key)

    return Func1(a, middle + 1, high, key);

  return Func1(a, low, middle, key);

}

Вторая функция реализует вставку на определенное место элемента в массив, со сдвигом всех элементов после него направо. То есть в конце массива создается новый элемент и все элементы, стоящие до вставляемого элемента, смещаются на один вправо. Таким образом, функция позволяет вставить новый элемент в любое место массива, с сохранением всех остальных элментов.