

1. (4 Punkte) Erstelle nach dem Verfahren aus dem Unterricht eine Tabelle für die Variablen L, R, mid und den Rückgabewert wenn die binäre Suche auf die Liste a und das gesuchte Element x angewendet wird.

```
def binaereSuche(a, x):  
    L = 0  
    R = len(a)-1  
    while L <= R:  
        mid = (L+R)//2  
        if a[mid] == x:  
            return mid  
        if a[mid] < x:  
            L = mid + 1  
        else:  
            R = mid - 1  
    return -1
```

x = 32

a = [4, 9, 13, 33, 35, 41, 46, 47, 49, 54, 58, 80, 89, 94, 97]

2. (4 Punkte) Erstelle nach dem Verfahren aus dem Unterricht eine Tabelle für die Variablen L, R, mid und den Rückgabewert wenn die binäre Suche auf die Liste a und das gesuchte Element x angewendet wird.

x = 49

a = [4, 7, 10, 11, 13, 17, 21, 24, 27, 29, 30, 32, 40, 45]

3. (4 Punkte) Erstelle nach dem Verfahren aus dem Unterricht eine Tabelle für die Variablen L, R, mid und den Rückgabewert wenn die binäre Suche auf die Liste a und das gesuchte Element x angewendet wird.

x = 13

a = [4, 7, 10, 11, 13, 17, 21, 24]

4. (4 Punkte) Die Liste a war ursprünglich sortiert. Dann wurde ein Anfangsteil nach hinten verschoben. Die Funktion rotatedList sucht mit binärer Suche den Index des alten Anfangs. Erstelle nach dem Verfahren aus dem Unterricht eine Tabelle für die Variablen L, R, mid, ans und den Rückgabewert.

```
def rotatedList(a):  
    L = 0  
    R = len(a)-1  
    ans = -1  
    while L <= R:  
        mid = (L+R)//2  
        if a[0] <= a[mid] :  
            ans = mid  
            L = mid + 1  
        else:  
            R = mid - 1  
    if ans == len(a)-1:  
        return -1  
    return ans+1
```

a = [21, 24, 27, 29, 30, 32, 40, 45, 4, 7, 10, 11, 13, 17]

5. (4 Punkte) Die Werte der Liste `a` sind erst aufsteigend, dann absteigend. Die Funktion `findMaximum` sucht mit binärer Suche den Index des größten Elements. Erstelle nach dem Verfahren aus dem Unterricht eine Tabelle für die Variablen `L`, `R`, `mid`, `ans` und den Rückgabewert.

```
def findMaximum(a):  
    L = 0  
    R = len(a)-1  
    ans = -1  
    while L <= R:  
        mid = (L+R)//2  
        if a[mid] > a[mid-1]:  
            ans = mid  
            L = mid + 1  
        else:  
            R = mid - 1  
    return ans
```

```
a = [21, 24, 27, 44, 6, 2]
```