

1 Aufgabe 1

```
static int binSearch(int[] a, int l, int r, int e)
{
    if(l<=r)
    {
        int m = ((l+r) / 2);
        if(a[m] > e)
            return binSearch(a,l,m-1,e);
        else if(a[m] < e)
            return binSearch(a,m+1,r,e);
        else if(a[m] == e)
            return m;
    }
    return -1;
}
```

2 Aufgabe 2

```
public static int[] alleVorkommen(int[] a, int e)
{
    int[] b = new int[a.length];
    int j = 0;
    for(int i =0;i<a.length;i++)
    {
        if(a[i] == e)
        {
            b[j] = i;
            j++;
        }
    }
    return b;
}
```

3 Aufgabe 3**4 Aufgabe 4**

```
public static int[][] negZahl(int[][] a)
{
    boolean neg = false;
```

```
int counter = 0;
int [][] b = new int[a.length][2];
marke:
for(int i=0;i<a.length;i++)
{
    neg = false;
    for(int j = 0;j<a[i].length;j++)
    {
        if(!neg)
        {
            if(a[i][j] < 0)
                neg = true; // Zahl < 0
        }
        else
        {
            if(a[i][j] < 0)
            {
                b[counter][0] = i;
                b[counter][1] = j;
                neg = false;
                counter++;
                continue marke;
            }
        }
    }
}
return b;
}
```

5 Aufgabe 5

```
enum Jahreszeit {Frühling, Sommer, Herbst, Winter}
public static boolean ist_warm(Jahreszeit j)
{
    switch(j)
    {
        case Frühling: return true;
        case Sommer: return true;
        default : return false;
    }
}
```