

1 Aufgabe 1

```
public static int kombination(int n, int k)
{
    return (int)(fakultaet(n) / (fakultaet(n-k) *
        fakultaet(k)));
}

private static int fakultaet(int n)
{
    if(n == 0)
        return 1;
    return (fakultaet(n-1) * n);
}
```

2 Aufgabe 2

```
public static int linSuche(double[] a, double w)
{
    for(int i = 0; i < a.length; i++)
    {
        if(a[i] == w)
            return i;
    }
    return -1;
}
```

3 Aufgabe 3

```
public static int[] laufSum(int[] a)
{
    if(a != null)
    {
        int summe = 0;
        int[] b = new int[a.length];
        for(int i = 0; i < a.length; i++)
        {
            summe = summe + a[i];
            b[i] = summe;
        }
        return b;
    }
}
```

```
    }  
    else return a;  
}
```

4 Aufgabe 4

```
public static int einfügen(int[] a, int r, int k)  
{  
    if(binSuche(a, k, 0, r) != -1)  
        return r;  
    else  
    {  
        for(int i=r; i>=0;i--)  
        {  
            if(a[i] > k)  
                a[i+1] = a[i];  
            if(a[i] < k)  
            {  
                a[i] = k;  
                break;  
            }  
        }  
        return r+1;  
    }  
}  
  
public static int binSuche(int[] a, int b, int u, int o)  
{  
    int m;  
    while(u<=o)  
    {  
        m = (u+o) / 2;  
        if(b > a[m])  
            u = m+1;  
        else if(b < a[m])  
            o = m-1;  
        else if (b == a[m])  
            return m;  
    }  
    return -1;  
}
```