

1a) 5,5/6

1. Prüfung, ob Spaltenanzahl von a der Zeilenanzahl von b entspricht. \rightarrow sonst Fehlermeldung
2. Array Ac Ergebnismatrix mit richtiger Größe anlegen.
3. Berechnung + Werte ablegen in Ergebnismatrix
4. Ergebnismatrix ausgeben,

1c) $O(n^3)$

Aufgabe1)

```
package uebung05;
```

```
public class Uebung051 {
```

```
    public static void main(String[] args) {
```

```
        // TODO Auto-generated method stub
```

```
        //Matrix A mit lxm Erstellen
```

```
        double[][] a = new double[][] {
```

```
            {2, 1, 7},
```

```
            {3, 0, 3},
```

```
        };
```

```
        // Matric'x B mit mxn erstellen
```

```
        double[][] b = new double[][] {
```

```
            {4, 2},
```

```
            {7, 1},
```

```
            {5, 2},
```

```
        };
```

```
        int spaltenanzahlA = a.length;
```

```
        int zeilenanzahlB = b[0].length;
```

Hier müsste man noch prüfen, ob es sich bei den Arrays tatsächlich um Matrizen handelt (alle Reihen müssen gleich viele Elemente enthalten)

```
        if (spaltenanzahlA == zeilenanzahlB) {
```

```

double[][] c = new double[spaltenanzahl][zeilenanzahl];

//zeile a
for(int i = 0; i < a.length; i++) {

    //spalte b
    for(int j = 0; j < b[i].length; j++) {

        // nummer
        for(int k = 0; k < b.length; k++) {

            c[i][j] += a[i][k] * b[k][j];

        }

    }

}

for(double[] vector : c) {
    for (double value : vector) {
        System.out.println(value);
    }
}

else {
    System.out.println("Eine Multiplikation kann nicht durchgeführt
werden.");
}

```

```
}
```

```
}
```

Aufgabe 2) 4,5/5

```
package uebung05;
```

```
public class Uebung052 {
```

```
    public static void main(String[] args) {
```

```
        // TODO Auto-generated method stub
```

```
    }
```

```
    //Aufgabe 2a)
```

```
    public static int min(int[] values) {
```

```
        selectionsort(values); Was ist, wenn das Array leer ist?
```

```
        return values[0];
```

```
    }
```

```
    public static int[] selectionsort(int[] resultarray) {
```

```
        for (int i = 0; i < resultarray.length; i++) {
```

```
            int minIndex = i;
```

```

        for(int j = i + 1; j < resultarray.length; j++) {
            if (resultarray[j] < resultarray[minIndex]) {
                minIndex = j;
            }
        }
        swap(resultarray, i, minIndex);
    }
    return resultarray;
}

```

```

private static void swap(int[] array1, int index1, int index2) {
    int tmp = array1[index1];
    array1[index1] = array1[index2];
    array1[index2] = tmp;
}

```

//Aufgabe 2b)

```

public static double product(double[] values) {
    double product = 1;
    for(double numbers: values) {
        product = numbers * product;
    }
    return product;
}

```

//Aufgabe 2c)

```
public static boolean isLatinLetter(char letter){  
    int l = letter;  
    if( l >= 65 && l <= 90) {  
  
        return true;  
  
    }  
  
    if( l >= 97 && l <= 122) {  
  
        return true;  
  
    }  
  
    else {  
        return false;  
    }  
}
```

//Aufgabe 2d)

```
public static int[][] createIdentityMatrix(int size){  
    int length = size;  
    int[][] ematrix = new int[length][length];  
    for( int i=0; i<ematrix.length; ++i) {  
        for( int j=0; j<ematrix.length; ++j) {  
            if( i == j) {
```

```

        ematrix[i][j] = 1;
    } else {
        ematrix[i][j] = 0;
    }

}

}

return ematrix;

}

}

```

Aufgabe 3 1,5/9

```

package uebung05;

public class Uebung053 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

    }

    public static int a(String array) {

        return array.length;
    }
}

```

```
}
```

```
        String      String  
public static int b(int[] keys, int key) {  
    int result = -1;  
    for(int i = 0; i < keys.length; i++) {  
        if (keys[i] == key) {  
            result = i;  
        }  
    }  
}
```

```
    return result;
```

```
}
```

```
}
```

11,5/20