



Achte auf eine übersichtliche und gut erläuterte Darstellung! Du brauchst keine Hilfsmittel! Bei der letzten Frage musst du nicht alle Teilaufgaben beantworten!!! **(Bearbeitungszeit: 60 Minuten)**

**0000. Aufgabe**

**(12 Punkte)**

- Implementiere einen beliebigen Algorithmus in Pseudocode (gerne auch C oder Java), der eine vorgegebene Liste von Zahlen der Größe nach absteigend sortiert.
- Angenommen, du sollst in einer Datenbank Kunden nach Nachnamen sortieren (von A-Z). Könntest du deinen Algorithmus aus a) „ummodelln“, um nicht nur Zahlen, sondern auch Namen zu sortieren? Erläutere nur das Prinzip!
- Was ist eigentlich eine „Datenbank“? Definiere kurz.

**0001. Aufgabe**

**(9 Punkte)**

Folgenden Codeschnipsel einer main-Methode in C++ haben wir im Unterricht besprochen:

```
char quit;  
quit = '\\0';  
while (quit != 'q')  
{  
    cout << "Hello ! This is a C++ program." << endl;  
    cout << "Press q to quit " << endl;  
    cin >> quit;  
}
```

- Erläutere, was dieser Code bewirkt.
- Was geschieht, wenn man eine Taste drückt?
- Vergleiche die Struktur einer while- mit der einer for-Schleife.
- Wofür ist die main-Methode in C-Programmen zuständig? Vergleiche mit Greenfoot.
- Was ist ein „char \*argv[]“ für ein Typ von Variable?
- Was ist ein Pointer? Welche Vorteile bietet dieser bei der Programmierung?

**0010. Aufgabe**

**(3 Punkte)**

Was ist ein Neuronales Netz in der Informatik und wieso bedient man sich manchmal neuronaler Netze anstelle von „einfachen“ Algorithmen?

## 0011. Aufgabe

(12-20 Punkte)

Beantworte nur zwei der Teilaufgaben a)-f)!

- a) Kryptographie: (6 Punkte)  
a. Was ist Kryptographie?  
b. Erläutere ein dir bekanntes Verschlüsselungsverfahren!
- b) Turing-Maschine: (8 Punkte)  
a. Was ist eine Turingmaschine?  
b. Erläutere die Addition von  $1+2=3$  auf einem Turingband.
- c) Arduino: (10 Punkte)  
a. Wofür sind „void setup()“ bzw. „void loop()“ bei Arduino da?  
b. Erläutere ausführlich, wofür das Arduino-Programm im Anhang gut ist und erkläre anhand dieses Programmes, wie ein Arduino-Programm grundsätzlich aufgebaut ist.
- d) Bildbearbeitung: (6 Punkte)  
a. Erläutere den Unterschied zwischen einer Vektor- und einer Rastergrafik.  
b. Was ist ein Histogramm?  
c. Erläutere das Rauschunterdrücken an einem Beispiel.
- e) Exotische Sprachen: (6 Punkte)  
a. Hier fehlt mir noch der Vortrag ;)
- f) C++: (10 Punkte)  
a. Erläutere ausführlich, was der Quellcode im Anhang bewirkt!  
b. Führe den Quellcode für die folgende Liste `list` bis zur zweiten Rekursion aus:

```
list [5] = {91,1,7,34,5} ;
```

## Zusatzaufgabe

(+3 Punkte)

Kennzeichne dazu eine der Teilaufgaben von 0011 mit einem (Z) und bearbeite sie vollständig!

## Anhang 1: Arduino Quellcode

```
int led_rot = 12;
int led_gelb = 11;
int led_gruen = 10;
int cnt = 0;
int state = 1;

void setup() {
  pinMode(led_rot, OUTPUT);
  pinMode(led_gelb, OUTPUT);
  pinMode(led_gruen, OUTPUT);
}

void loop() {
  cnt++;
  if(cnt == 100) {
    cnt = 0;
    statemaschine();
  }

  delay(10);
}

void statemaschine() {
  switch(state) {
    case 1:
      digitalWrite(led_rot, HIGH);
      digitalWrite(led_gelb, LOW);
      digitalWrite(led_gruen, LOW);
      state++;
      break;

    case 2:
      digitalWrite(led_rot, HIGH);
      digitalWrite(led_gelb, HIGH);
      digitalWrite(led_gruen, LOW);
      state++;
      break;

    case 3:
      digitalWrite(led_rot, LOW);
      digitalWrite(led_gelb, LOW);
      digitalWrite(led_gruen, HIGH);
      state++;
      break;

    case 4:
      digitalWrite(led_rot, LOW);
      digitalWrite(led_gelb, HIGH);
      digitalWrite(led_gruen, LOW);
      state = 1;
      delay(100);
      break;
  }
}
```

## Anhang 2: C++ Quellcode

```
int* quickSort (int* arr, int left, int right)
{
    int i= left, j=right;
    int temp;
    int pivot = arr[(left+right)/2];

    while (i<=j)
    {
        while (arr[i]<pivot){
            i++;
        }
        while (arr[j]>pivot){
            j--;
        }

        if (i<=j)
        {
            temp = arr[i];
            arr[i]=arr[j];
            arr[j]=temp;
            i++;
            j--;
        }
    }

    if (left<j)
        quickSort(arr, left, j);
    if (i<right)
        quickSort (arr,i,right);

    return arr;
}
```