Klausur 1 – 22.01.2018

(90 Min., keine Hilfsmittel
außer einfachem, nicht-programmierbarem Taschenrechner)

Name, Vorname: [redacted] Matrikelnr.: [redacted]

[letzte Prüfungsmöglichkeit wg. (zutreffenden falls ankreuzen): ❑ 3. Versuch]

1. Aufgabe — 2 /2
   Angenommen, bei einem zukünftigen Quantencomputer könnte jede Speicherstelle (*qubit*) vier verschiedene Zustände annehmen und ein byte bestünde aus 8 solchen *qubit*s. Welches wäre dann die mit einem solchen byte darstellbare größte, vorzeichenlose Ganzzahl (im Dezimalsystem)?
   Begründen Sie Ihre Angabe.

2. Aufgabe
   a) Erläutern Sie den Unterschied der Parameterübergabe zwischen den beiden Funktionsaufrufen — 4 /4

```
double e, arg = 2.;
unsigned int r, par = 3;
e = func1(&arg);
r = func2(par);
```

   und wozu dies nützlich ist.

   b) Notieren Sie die jeweils passenden Prototypen beider Funktionen. — 2 /2

3. Aufgabe — 7 /7
   Geben Sie die formatierte Ausgabe des folgenden Programms an:

```
#include <stdio.h>
int main() {
      int i = 0, n = 33;
      while (n > 2) {
            if (n % 2)
                  n = n * 2 - 2;
            else
                  n /= 2 + 1;
            i++;
            if (n % 10 == 0) continue;
            printf("%2d: %2d\n",i,n);
      }
      return 0;
}
```

4. Aufgabe — 2 /4
   Geben Sie an, was und warum an folgender Struktur-Definition fehlerhaft ist und notieren Sie eine korrekte Version:

```
typedef struct {
      int ID;
      *char name;
      char text[];
} Data;
```

5. Aufgabe — 5,5 /6
   Schreiben Sie eine Funktion, die als Parameter ein eindimensionales Feld von positiven Ganzzahlen und seine (belegte) Länge erhält. Ihre Funktion soll die *Summe* der Elemente im Feld zurückgeben, die das Dreifache ihres Vorgänger-Elementes betragen.

Übertrag: 22,5 /25

Übertrag: 22,5 /25

6. Aufgabe

a) Schreiben Sie Strukturen zur Definition eines einfachen `Termin`-Datentyps. Ein `Termin` soll die Attribute `Bezeichnung`, `Ort` und `Zeit` besitzen, die mit geeigneten Datentypen zu definieren sind. Insbesondere soll `Zeit` selbst wiederum vom Strukturtyp `Zeitpunkt` sein, der aus den Komponenten `Tag`, `Monat`, `Jahr`, `Stunde` und `Minute` besteht, die jeweils mit einem geeigneten Typ zu definieren sind. 6 /6

b) Schreiben Sie eine Funktion `termin_offen`, mit der überprüft werden kann, ob der ihr übergebene `Termin` noch stattfinden wird. Gehen Sie dazu davon aus, dass Sie eine vordefinierte Funktion now aufrufen können, deren Prototyp

```
Zeitpunkt now(void);
```

lautet und die Ihnen die aktuelle Zeit zurück gibt. 6 /10

7. Aufgabe 14 /16

Gegeben seien folgende Definitionen:

```
enum { MAXBTL = 1000 };
typedef enum { Stahl, Messing, Aluminium, Kupfer } werkstoff_t;
typedef struct {
      unsigned int ID;
      char *Bezeichnung;
      werkstoff_t Material;
      float Stueckpreis;
} bauteil_t;
bauteil_t Bauteile[MAXBTL];
```

Schreiben Sie eine Funktion, die das `Bauteile`-array, seine (belegte) Länge und einen `ReferenzPreis` erhält (alle Werte werden als sinnvoll vorbelegt vorausgesetzt). Die Funktion soll die *Anzahl* aller `Messing-Bauteile` zurückliefern, deren `Stueckpreis` unterhalb des `ReferenzPreises` liegt und den *Mittelwert* ihrer `Stueckpreise`.

1,7

48,5 /57

29.01.2018, Ra

Klausur IN1 – TIB 22.01.2018

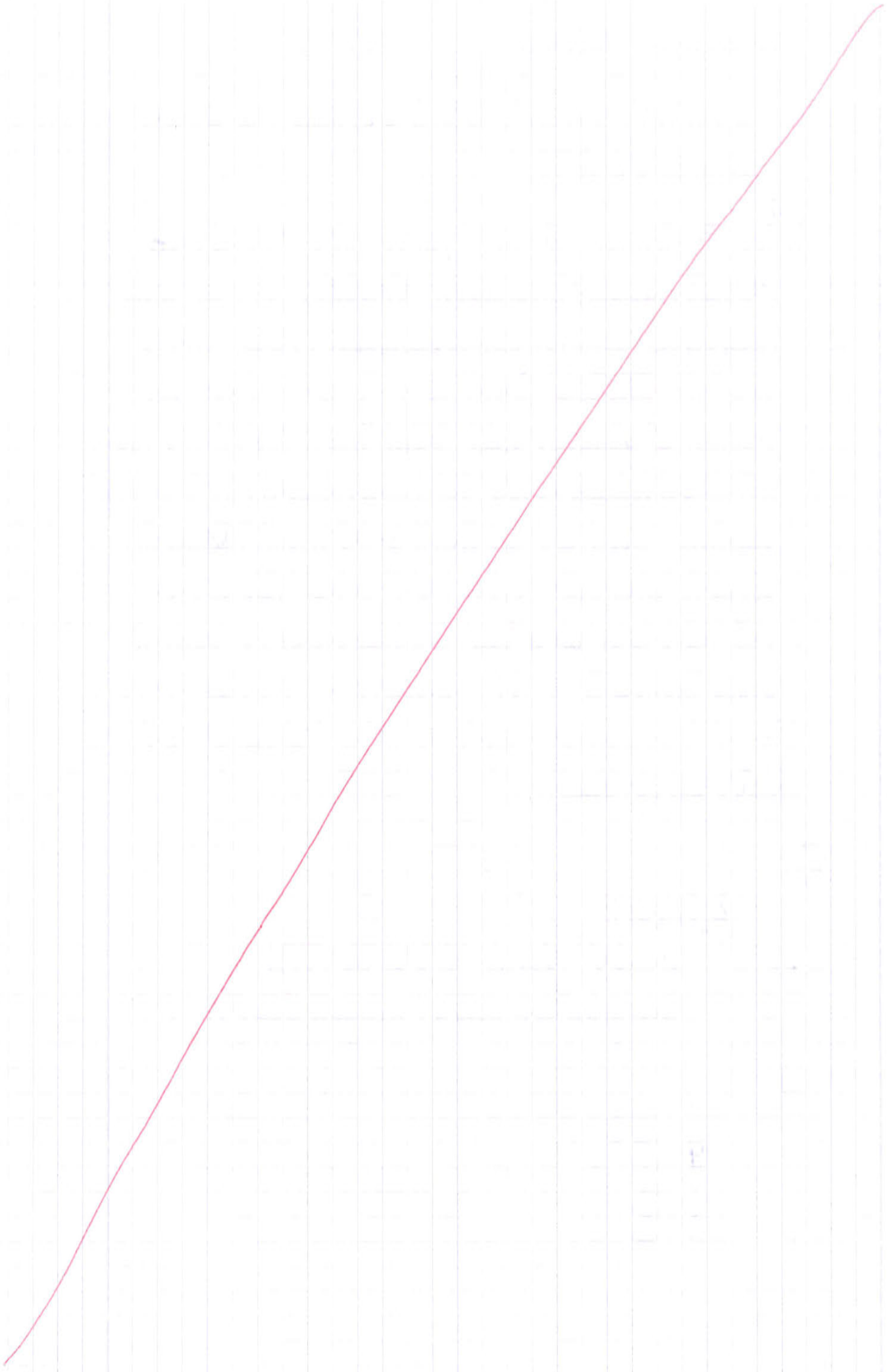
[redacted] [redacted]

1) 65535, da $4^8 = 65536$, dies die 0 aber ausschließt.

2) a) Der Funktion func1 wird die Adresse des Aktualparameters übergeben. Erkennbar ist dies am Ampersand vor den Variablennamen, dem Adressoperator. Dieser Vorgang wird call-by-reference genannt und bietet die Möglichkeit, den Variableninhalt direkt zu manipulieren, anders als in func2. Der hier stattfindende Vorgang wird als call-by-value bezeichnet. Hier wird der Wert des Aktualparameters kopiert und sämtliche Manipulationen innerhalb der Funktionen finden an der Kopie statt.

2b)
```
double func1( double* x);
unsigned int func2( unsigned int x);
```

3)
```
1: 64
2: 21
4: 13
5: 24
6:  8
7:  2
```



Klausur JV1 - TIB 22.01.2018

[redacted] [redacted]

4) Der Fehler liegt in der Deklaration des Zeigers name vom Typ char. Die dargestellte Schreibweise führt zu einen Compiler-Fehler. Die korrekte Deklaration wäre "char* name;"

f: 2. Fehler nicht erkannt: char text[2=0];
f: Notation korrekte struct fehlt?

5)

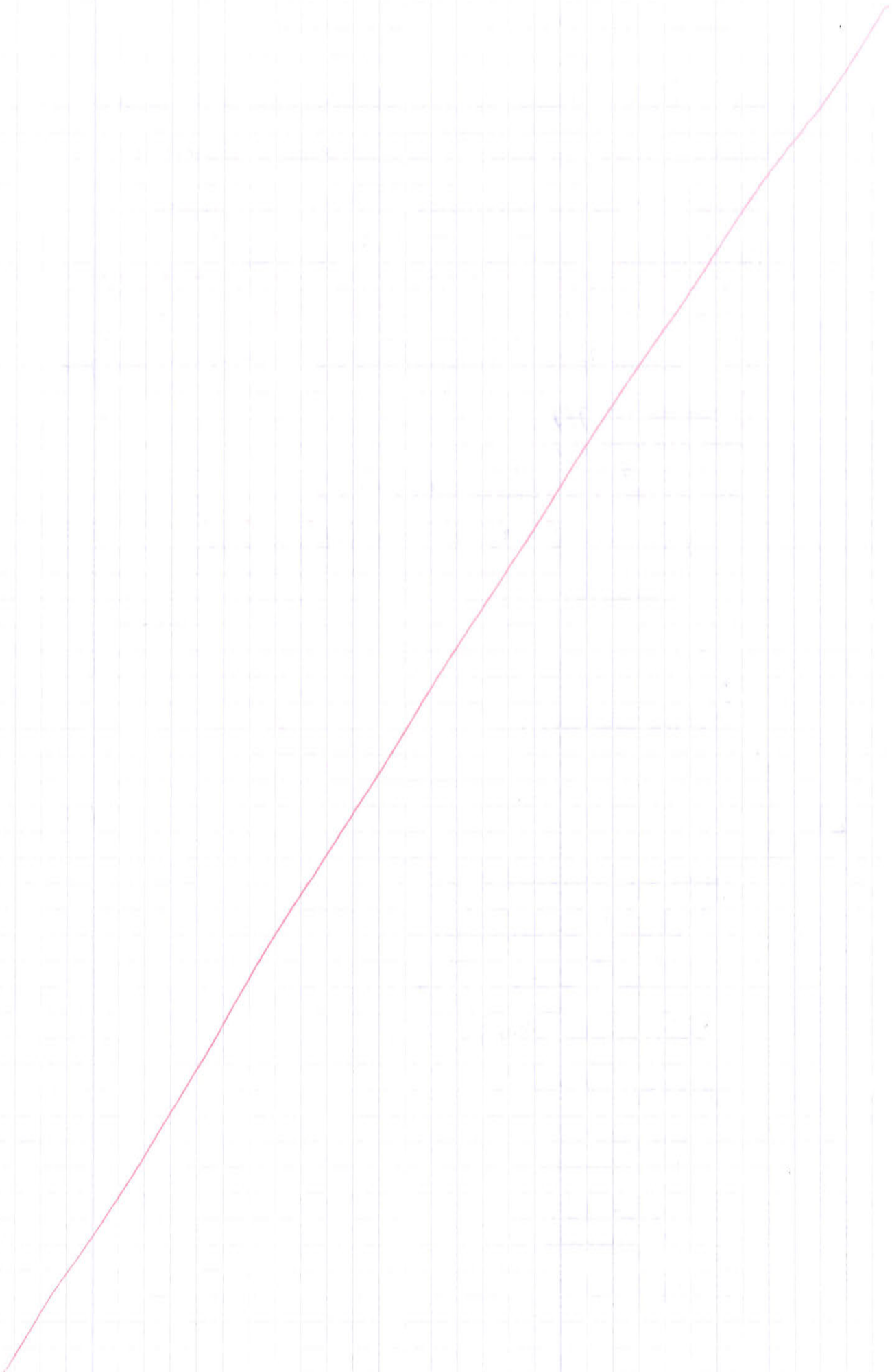
```
int summe3fach (unsigned int[] zahlen, int laenge){
(int zaehler, i)
int zaehler, i;
for (i=0; i< laenge; i++){
  if (zahlen[i] == zahlen[i-1] * 3){
    zaehler++;
  }
}
return zaehler;
}
```

(const)

6)

a)

```
typedef struct {
  char* Bezeichnung;
  char* Ort;
  Zeitpunkt Zeit;
} typedef struct {
  int Jahr;
  int Monat;
  int Tag;
  int Stunde;
  int Minute;
} Zeitpunkt;
```



Klausur JN1 – TIB 22.01.2018

[redacted] [redacted]

noch 6 a)

```
typedef struct {
    Zeitpunkt Zeit;
    char* Ort;
    char* Bezeichnung;
} Termin;
```

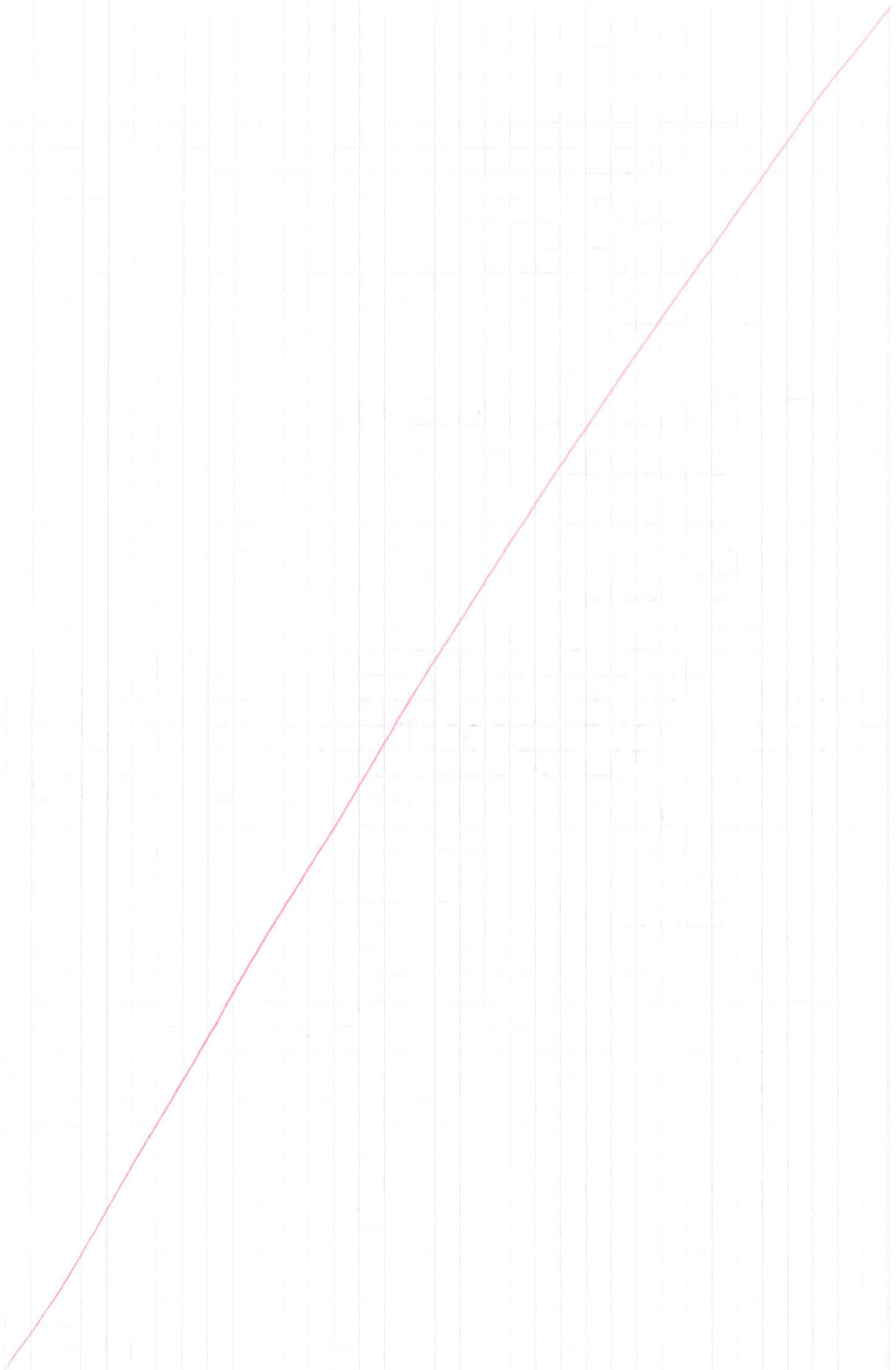
b)

```
int termin_offen (Termin t) {
    Zeitpunkt z;
    int boo = 0;
    z = now();
    if (t.Zeit.Jahr ≥ z.Jahr) {
        if (t.Zeit.Monat ≥ z.Monat) {
            if (t.Zeit.Tag ≥ z.Tag) {
                if (t.Zeit.Stunde ≥ z.Stunde) {
                    if (t.Minute ≥ z.Minute) {
                        boo = 1;
                    }
                }
            }
        }
    }
    return boo;
```

Das ist eine große && -Verknüpfung!

Logik f: wenn z.B. das Jahr des Termins größer als das aktuelle ist, darf der Monat gar nicht mehr verglichen werden!

3 v. 4



Klausur - IN1 - TIB 22.01.2018

[redacted]

7) typedef struct {
    float MittelPreis;
    int AnzahlBauteile;
} Rueckgabe;

Rueckgabe suche(bauteil_t teile[], int laenge,
                float RefPreis) {
    float summe = 0.0;
    int zaehler = 0;
    Rueckgabe ret;
    for (i=0; i<laenge; i++) {
        if (teile[i].material == Messing && teile[i].Stueckpreis < RefPreis) {
            summe += teile[i].Stueckpreis;
            zaehler++;
        }
    }
    if (zaehler > 0)
    ret.Mittelpreis = summe / (float) zaehler;
    ret.AnzahlBauteile = zaehler;
    return ret;
}

ist direkt die Konstante.

fehlt

unnötig