
Übungsblatt 5

Abgabe spätestens bis: 07.12.2020 10:00 Uhr

- Dieses Übungsblatt soll in den in der Übungsgruppe festgelegten Teams abgegeben werden (Einzelabgaben sind erlaubt, falls noch keine Teamzuteilung erfolgt ist).
- Die **Zeitangaben** geben zur Orientierung an, wie viel Zeit für eine Aufgabe später in der Klausur vorgesehen wäre; gehen Sie davon aus, dass Sie zum jetzigen Zeitpunkt wesentlich länger brauchen und die angegebene Zeit erst nach ausreichender Übung erreichen.

* leichte Aufgabe / ** mittelschwere Aufgabe / *** schwere Aufgabe

Aufgabe 17 * (*Gleitkomma-Codierung und -Dekodierung, 26 Minuten*)

Geben Sie zu jeder Teilaufgabe jeweils den Rechenweg oder eine Begründung an.

a) (Codierung des Exponenten, 12 Minuten)

1. (*, 1 Minute)
Berechnen Sie: $c_{EX-127,8}(2)$
2. (*, 1 Minute)
Berechnen Sie: $c_{EX-127,8}(-2)$
3. (*, 1 Minute)
Berechnen Sie: $(10010100)_{EX-127,8}$
4. (*, 1 Minute)
Berechnen Sie: $(00001001)_{EX-127,8}$
5. (*, 1 Minute)
Welche $EX-q$ -Codierung wird zur Codierung des Exponenten in der Gleitkomma-Codierung $c_{GK,53,64}$ verwendet?
6. (*, 1 Minute)
Welche $EX-q$ -Codierung wird zur Codierung des Exponenten in der Gleitkomma-Codierung $c_{GK,11,16}$ verwendet?
7. (*, 1 Minute)
Geben Sie das Bitmuster des größten darstellbaren Exponenten der Gleitkomma-Codierung $c_{GK,53,64}$ an (unter Berücksichtigung von reservierten Bitmustern).
8. (*, 1 Minute)
Geben Sie das Bitmuster des kleinsten darstellbaren Exponenten der Gleitkomma-Codierung $c_{GK,11,16}$ an (unter Berücksichtigung von reservierten Bitmustern).

-
9. (**, 1 Minute)
Geben Sie den größten darstellbaren Exponenten der Gleitkomma-Codierung $c_{GK,11,16}$ als Dezimalzahl an (unter Berücksichtigung von reservierten Bitmustern).
 10. (**, 1 Minute)
Geben Sie den kleinsten darstellbaren Exponenten der Gleitkomma-Codierung $c_{GK,24,32}$ als Dezimalzahl an (unter Berücksichtigung von reservierten Bitmustern).
 11. (**, 1 Minute)
Führen Sie folgende Addition von Bitmustern in der $c_{EX-127,8}$ -Codierung durch:
 $10000000 \oplus_{EX-127,8} 00000000$
 12. (**, 1 Minute)
Führen Sie folgende Subtraktion von Bitmustern in der $c_{EX-127,8}$ -Codierung durch:
 $01100000 \ominus_{EX-127,8} 00100000$
- b) (Codierung der Mantisse, 12 Minuten)
1. (*, 1 Minute)
Berechnen Sie: $c_{FK,23,23}(0.25)$
 2. (*, 1 Minute)
Berechnen Sie: $c_{FK,10,10}(0.1)$
 3. (*, 1 Minute)
Berechnen Sie: $(101000000000000000000000)_{FK,23,23}$
 4. (*, 1 Minute)
Berechnen Sie: $(0000000001)_{FK,10,10}$
 5. (*, 1 Minute)
Welche Festkomma-Codierung wird zur Codierung der Mantisse in der Gleitkomma-Codierung $c_{GK,11,16}$ verwendet?
 6. (*, 1 Minute)
Welche Festkomma-Codierung wird zur Codierung der Mantisse in der Gleitkomma-Codierung $c_{GK,65,80}$ verwendet?
 7. (*, 1 Minute)
Wie groß ist maximal der absolute Rundungsfehler bei der $c_{FK,10,10}$ -Codierung?
 8. (*, 1 Minute)
Wie viele Bit k für Nachkommastellen muss eine Festkomma-Codierung mindestens haben, damit der absolute Rundungsfehler maximal 0.02 ist?
 9. (**, 2 Minuten)
Berechnen Sie den absoluten Rundungsfehler bei der Codierung $c_{FK,23,23}(0.25)$.
 10. (**, 2 Minuten)
Berechnen Sie den absoluten Rundungsfehler bei der Codierung $c_{FK,4,4}(0.275)$.
- c) (Normierung, 2 Minuten)
1. (*, 1 Minute)
Geben Sie die Dezimalzahl 16.8 in der Form $m \cdot 2^e$ mit $1 \leq m < 2$ an.
 2. (*, 1 Minute)
Geben Sie die Dezimalzahl 0.01 in der Form $m \cdot 2^e$ mit $1 \leq m < 2$ an.

Aufgabe 18 (Gleitkomma-Codierung und -Dekodierung, 31 Minuten)

1. (**, 4 Minuten)
Berechnen Sie : $c_{GK,11,16}(4.8)$
2. (**, 4 Minuten)
Berechnen Sie: $c_{GK,11,16}(-0.525)$
3. (*, 2 Minuten)
Berechnen Sie: $(1100111001000000)_{GK,11,16}$
4. (*, 2 Minuten)
Berechnen Sie: $(01000010000001100000000000000000)_{GK,24,32}$
5. (*, 1 Minute)
Geben Sie die größte positive, normalisiert darstellbare Zahl in der Gleitkomma-Codierung $c_{GK,11,16}$ in normierter Gleitkommadarstellung zur Basis 2 an (unter Berücksichtigung von reservierten Bitmustern).
6. (*, 1 Minute)
Geben Sie die kleinste positive, normalisiert darstellbare Zahl in der Gleitkomma-Codierung $c_{GK,11,16}$ in normierter Gleitkommadarstellung zur Basis 2 an (unter Berücksichtigung von reservierten Bitmustern).
7. (*, 1 Minute)
Wie groß ist maximal der relative Rundungsfehler bei der $c_{GK,11,16}$ -Codierung?
8. (**, 1 Minute)
Wie groß muss man k mindestens wählen, damit der **relative** Rundungsfehler bei der Codierung einer Zahl mit einer Gleitkommacodierung $c_{GK,k,n}$ maximal 0.01 ist?
9. (***, 1 Minute)
Wie groß muss man k mindestens wählen, damit bei der $c_{GK,k,n}$ -Codierung einer positiven Zahl x mit einer normierten Gleitkommadarstellung $x = m \cdot 2^{10}$ ein maximaler **absoluter** Rundungsfehler von 0.01 auftritt?
10. (**, 4 Minuten)
Berechnen Sie den absoluten Rundungsfehler bei der Codierung $c_{GK,11,16}(5.0)$.
11. (**, 4 Minuten)
Berechnen Sie den absoluten Rundungsfehler bei der Codierung $c_{GK,5,8}(14.4)$.
12. (**, 2 Minuten)
Berechnen Sie das Ergebnis der Addition der Zahlen $(1.0)_2 \cdot 2^0$ und $(1.1)_2 \cdot 2^{-1}$ bzgl. der $c_{GK,5,8}$ -Codierung und geben Sie den dabei ggf. auftretenden Rundungsfehler an.
13. (**, 2 Minuten)
Berechnen Sie das Ergebnis der Addition der Zahlen $(1.0)_2 \cdot 2^0$ und $(1.1)_2 \cdot 2^0$ bzgl. der $c_{GK,5,8}$ -Codierung und geben Sie den dabei ggf. auftretenden Rundungsfehler an.
14. (**, 2 Minuten)
Berechnen Sie das Ergebnis der Addition der Zahlen $(1.0)_2 \cdot 2^3$ und $(1.0)_2 \cdot 2^{-2}$ bzgl. der $c_{GK,5,8}$ -Codierung und geben Sie den dabei ggf. auftretenden Rundungsfehler an.

Aufgabe 19 (*C-Ausdrücke, 20 Minuten*)

1. (*, 0,5 Minuten) Welchen Wert hat der Ausdruck `!1.05`?
2. (*, 0,5 Minuten) Welchen Wert hat der Ausdruck `!\0`?
3. (*, 0,5 Minuten) Welchen Wert hat der Ausdruck `-1 < 0`?
4. (*, 0,5 Minuten) Welchen Wert hat der Ausdruck `!(-1 < 0)`?
5. (*, 0,5 Minuten) Welchen Wert hat der Ausdruck `(0 < 1) && (1 < 0)`?
6. (*, 0,5 Minuten) Welchen Wert hat der Ausdruck `(0 < 1) || (1 < 0)`?
7. (*, 0,5 Minuten) Welchen Wert hat der Ausdruck `!1 || 0`?
8. (*, 0,5 Minuten) Welchen Wert hat der Ausdruck `!(1 || 0)`?
9. (*, 0,5 Minuten) Welchen Wert hat der Ausdruck `!(1 && 0)`?
10. (*, 0,5 Minuten) Welchen Wert hat der Ausdruck `'B' != 66`?
11. (*, 0,5 Minuten) Welchen Wert hat der Ausdruck `'A' == 'a'`?
12. (*, 0,5 Minuten) Welchen Wert hat der folgende Ausdruck: `1 ? 2 : 3`
13. (*, 0,5 Minuten) Welchen Wert hat der folgende Ausdruck: `1 ? (2 ? 3 : 4) : 5`
14. (*, 0,5 Minuten) Welchen Wert hat der Ausdruck `isdigit(65)`?
15. (*, 0,5 Minuten) Welchen Wert hat der Ausdruck `isupper(65)`?
16. (*, 0,5 Minuten) Welchen Wert hat der Ausdruck `y = 'A'`?
17. (*, 0,5 Minuten) Sei `y` eine `int`-Variable mit dem Wert 0. Welchen Wert hat der Ausdruck `y += -1`?
18. (*, 0,5 Minuten) Sei `y` eine `int`-Variable mit dem Wert 5. Welchen Wert hat der Ausdruck `y *= -1`?
19. (*, 0,5 Minuten) Sei `y` eine `int`-Variable mit dem Wert 5. Welchen Wert hat der Ausdruck `y /= 2`?
20. (*, 0,5 Minuten) Sei `y` eine `double`-Variable mit dem Wert 5.0. Welchen Wert hat der Ausdruck `y /= 2`?
21. (*, 0,5 Minuten) Sei `y` eine `int`-Variable mit dem Wert 0. Welchen Wert hat der Ausdruck `++y`?
22. (*, 0,5 Minuten) Sei `y` eine `int`-Variable mit dem Wert 0. Welchen Wert hat der Ausdruck `y++`?
23. (*, 0,5 Minuten) Sei `y` eine `int`-Variable mit dem Wert 0. Welchen Wert hat der Ausdruck `y == 2`?
24. (*, 0,5 Minuten) Sei `y` eine `int`-Variable mit dem Wert 0. Welchen Wert hat die Variable `y` nach Auswertung des Ausdrucks `--y`?
25. (*, 0,5 Minuten) Sei `y` eine `int`-Variable mit dem Wert 0. Welchen Wert hat die Variable `y` nach Auswertung des Ausdrucks `y--`?
26. (*, 0,5 Minuten) Welchen Wert hat der Ausdruck `x = y = 8`?
27. (*, 0,5 Minuten) Sei `y` eine `int`-Variable mit dem Wert 0. Welchen Wert hat der Ausdruck `y == (x = 0)`?

-
28. (*, 0,5 Minuten) Welchen Wert hat die Variable `x` nach Auswertung des Ausdrucks
`x = y = -2`?
29. (**, 1 Minute) Seien `x` und `y` Variablen vom Typ `int`. Formulieren Sie eine C-Bedingung, die genau dann wahr ist, wenn `x` durch `y` teilbar ist.
30. (**, 1 Minute) Sei `v` ein Feld vom Typ `int`. Formulieren Sie eine C-Bedingung, die genau dann wahr ist, wenn der Speicherbereich von `v` mindestens für 10 Komponenten ausreicht.
31. (**, 1 Minute) Seien `x` und `k` Variablen vom Typ `int`. Formulieren Sie eine C-Bedingung mit bitweisem Operator, die genau dann wahr ist, wenn `x` echt kleiner 2^k ist.
32. (**, 1 Minute) Seien `x` und `y` Variablen vom Typ `double`. Formulieren Sie einen C-Ausdruck, dessen Wert das Minimum von `x` und `y` ist.
33. (**, 1 Minute) Sei `x` eine Variable vom Typ `double`. Formulieren Sie einen C-Ausdruck, dessen Wert der zu `x` nächstgelegene ganzzahlige Wert ist. Benutzen Sie dazu geeignete mathematische Funktionen aus `math.h`.

Aufgabe 20 (C-Kontrollstrukturen, 24 Minuten)

1. (*, 3 Minuten)
Drücken Sie die folgende Fallunterscheidung **ohne** Benutzung von `switch-case` aus:

```
switch (alter / 10) {
case 0:
    printf("under 10\n");
    break;
case 1:
    printf("between 10 and 20\n");
    break;
default:
    printf("over 20\n");
    break;
}
```

2. (*, 3 Minuten)
Drücken Sie den folgende Fallunterscheidung **mit** Benutzung von `switch-case` aus:

```
if (c == 'e') {
    printf("%e\n", x);
} else if (c == 'f') {
    printf("%f\n", x);
} else {
    printf("invalid parameter\n");
}
```

3. (*, 3 Minuten)
Drücken Sie die folgende `do`-Schleife mit einer `while`-Schleife aus:

```
i = 0;
do {
    ++i;
    printf("%i\n", rand());
} while (i <= x);
```

4. (*, 3 Minuten)
Drücken Sie die folgende `while`-Schleife mit einer `do`-Schleife aus:

```
summe = v[0];
i = 1;
while (i <= n) {
    summe += v[i];
    ++i;
}
```

5. (*, 3 Minuten)

Drücken Sie die folgende **do**-Schleife mit einer **do**-Schleife mit Schleifenbedingung 1 aus:

```
summe = 0;
i = 0;
do {
    summe += v[i];
    ++i;
} while (i <= n);
```

6. (*, 3 Minuten)

Drücken Sie die folgende **while**-Schleife mit einer **while**-Schleife ohne **break**-Anweisung aus:

```
s = 0;
while (1) {
    if (x == 0) {
        break;
    }
    x = x / 2;
    ++s;
}
```

7. (*, 3 Minuten)

Drücken Sie die folgende **while**-Schleife mit einer **while**-Schleife mit **continue**-Anweisung aus:

```
i = 0;
while (i < n) {
    if ((x = rand()) % 2 != 0) {
        printf("%i\n", x);
        ++i;
    }
}
```

8. (*, 3 Minuten)

Drücken Sie die folgende Schleife mit einer **while**-Schleife ohne **continue**-Anweisung aus:

```
s = 0;
while (x > 0) {
    r = x % 2;
    x = x / 2;
    if (r != 0) {
        continue;
    }
    ++s;
}
```