Hilfsmittel

Unicode-Masken

```
Unicode(hex)  Maske
bis 7F        0xxxxxxx
bis 7FF       110xxxxx 10xxxxxx
bis FFFF      1110xxxx 10xxxxxx 10xxxxxx
bis 10FFFF    11110xxx 10xxxxxx 10xxxxxx 10xxxxxx
```

Zeichen und die dezimalen ASCII-Werte

```
chr:      !   "   #   $   %   &   '   (   )   *   +   ,   -   .   /
asc: 32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47
chr:  0   1   2   3   4   5   6   7   8   9   :   ;   <   =   >   ?
asc: 48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63
chr:  @   A   B   C   D   E   F   G   H   I   J   K   L   M   N   O
asc: 64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79
chr:  P   Q   R   S   T   U   V   W   X   Y   Z   [   \   ]   ^   _
asc: 80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95
chr:  `   a   b   c   d   e   f   g   h   i   j   k   l   m   n   o
asc: 96  97  98  99  100 101 102 103 104 105 106 107 108 109 110 111
chr:  p   q   r   s   t   u   v   w   x   y   z   {   |   }   ~
asc: 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127
```

Weitere Zeichen mit ihrem dezimalen Unicode-Codepoint:

```
ä 228 - ö 246 - ü 252 - Ä 196 - Ö 214 - Ü 220 - ß 223 - € 8364
alpha α 945 - beta β 946 - gamma γ 947 - delta δ 948 - epsilon ε 949
pik ♠ 9824 - herz ♡ 9825 - karo ♢ 9826 - kreuz ♣ 9827
```

BOM für UTF-8 Codierung: EFBBBF

1. (2 Punkte) ANSI ist eine Erweiterung des ASCII-Codes auf 8 Bit. Im ANSI-Code hat 'a' die Codierung 01100001, 'A' die Codierung 01000001. Dekodiere die ANSI-Code Bitfolge:

   `01010110 01101001 01100101 01101100`

   Lösung: Viel

2. (3 Punkte) Das mathematische Summenzeichen hat den dezimalen Unicode Code Point 8721. Gib die UTF-8 Codierung in hexadezimaler Schreibweise an.

   Lösung:

```
        Codepoint dezimal : 8721
    Codepoint hexadezimal : 2211
         Binärdarstellung : 0010 0010 0001 0001
                    Maske : 1110 xxxx 10xx xxxx 10xx xxxx
              eingesetzt  : 1110 0010 1000 1000 1001 0001
            UTF-Codierung : E28891
```

3. (3 Punkte) Ein Texteditor stellt folgende Bitfolge dar:

```
1100 0011 1011 0110 0100 0001 1100 1110 1011 0011 0111 0111
```

Gib die hexadezimalen Codepoints der dargestellten Zeichen an. Prüfe, ob die Zeichen in der Hilfsmitteltabelle auftauchen und gib dann an, um welche Zeichen es sich handelt.

Lösung:

```
1100 0011 1011 0110 0100 0001 1100 1110 1011 0011 0111 0111
Die Bitfolge enthält 4 Zeichen:
Zeichen 1:  codepoint dezimal = 246, codepoint hex = f6, Zeichen ö
Zeichen 2:  codepoint dezimal = 65, codepoint hex = 41, Zeichen A
Zeichen 3:  codepoint dezimal = 947, codepoint hex = 3b3, Zeichen gamma
Zeichen 4:  codepoint dezimal = 119, codepoint hex = 77, Zeichen w
```

4. (3 Punkte) a. Was sagt die Fano-Bedinung aus?

b. Unten sind Codewörter für einige Buchstaben gegeben. Ist die Fano-Bedingung erfüllt? Finde ein Beispiel für eine Codierung, die nicht eindeutig dekodiert werden kann.

a = '00' b = '10' c = '11' d = '01' f = '111' g = '100'

Lösung: a. Die Fanobedingung besagt, dass kein Codewort der Anfang eines anderen Codewortes sein darf. Wenn die Fanobedingung erfüllt ist, lassen sich Codierungen eindeutig decodieren.

b. Das Codewort für c ist der Anfang des Codeworts für f. Die Fano-Bedingung ist nicht erfüllt. Beispiel für eine nicht eindeutige Codierung: 11101100 kann mit cbca oder fdg dekodiert werden.

5. (3 Punkte) Decodiere die Bitfolge mit dem angegebenen Huffman-Baum. Gib für jedes Zeichen die Codierung an.

```
11010011111100011100001101
```

Hinweis zur Darstellung des Huffman-Baums: Der Baum ergibt sich durch Drehen der abgebildeten Zeilen um 90 Grad nach rechts. Die Punkte geben die Ebenen des Baumes an. Knoten, die nur Zahlen aber keine Zeichen enthalten, sind mit einem # markiert.

```
...2 t
..4 #
...2 b
.6 #
...1 g
..2 #
...1 u
10 #
..2 r
.4 #
..2 e
```

Lösung:

```
b->110, u->100, t->111, e->00, r->01, g->101
Kodiertes Wort: butterberg
```

6. (4 Punkte) Konstruiere nach dem Verfahren aus dem Unterricht einen Huffman-Baum für das Wort `innenminister`. Gib für jeden Buchstaben die Codierung an.

Lösung:

```
Huffmanbaum für: innenminister
....1 s
...2 #
....1 m
..4 #
...2 e
.8 #
..4 n
13 #
..3 i
.5 #
...1 r
..2 #
...1 t

Codierung:
i -> 01
n -> 10
e -> 110
m -> 1110
s -> 1111
t -> 000
r -> 001
```

7. (3 Punkte) a. Benenne im Beispiel: globale Variable, lokale Variable, Parameter und Argument.
b. Was wird ausgegeben?

```
def calc(z):
    a = 6
    return a*z

y = 3
w = calc(4)+y
print(w)
```

Lösung:

a. y, w sind globale Variablen, a, z sind lokale Variablen, z ist Parameter, 4 ist Argument.
b. 27

8. (1 Punkt) Schreibe eine Funktion, die sich so verhält, wie im folgenden docstring beschrieben.

```
'''
returns: string, die Zeichenfolge 'lol'

>>> go()
'lol'
'''
```

Lösung:

```
def go():
    return 'lol'
```

9. (2 Punkte) Schreibe eine Funktion, die sich so verhält, wie im folgenden docstring beschrieben.

```
'''
s string, k >= 0, k optional mit default 1
returns: string, die ersten k zeichen von s

>>> make('Hallo')
'H'
>>> make('Hallo',3)
'Hal'
'''
```

Lösung:

```
def make(s, k=1):
    return s[:k]
```

10. (4 Punkte) Was erscheint auf der Konsole?

```
def a(x):
    return x - 2
def b(x):
    return x + 4
def c(x,y):
    return x + 2*y
def d(x,y):
    return x > y
z = c(a(2),b(6))
print(d(z,8),z)
```

Lösung: True 20

11. (1 Punkt) Was wird ausgegeben?

```
s = 'Ha'
s2 = 'H'+'a'
print(s is s2, s == s2)
```

Lösung: True True

12. (2 Punkte) Übersetze den folgenden Pseudocode in Python-Code:

```
Falls der Variablen best kein Wert zugeordnet ist oder x größer als best ist:
    setze best auf x
```

Lösung:

```
if best is None or x > best:
    best = x
```

13. (1 Punkt) Was wird ausgegeben?

```
a = [9]
b = a
a.append(4)
b.append(2)
print(a, b)
```

Lösung:

```
[9, 4, 2] [9, 4, 2]
```

14. (2 Punkte) Was wird ausgegeben?

```
def doit(x):
    x.append(12)
    a = [7]
    return x

a = [4]
b = doit(a)
print(a, b)
```

Lösung:

```
[4, 12] [4, 12]
```

15. (3 Punkte) Berechne die Datenmenge in KByte eines Bildes, das 4.2 cm breit und 5.6 cm hoch ist und eine Bildauflösung von 80 dpi hat (1 inch = 2.54 cm).
a. als Schwarzweiß-Bild.
b. als Graustufen-Bild mit 16 Farben.
c. als Farbbild.

Lösung:

```
Bildgröße: 4.2 cm * 5.6 cm = 1.65 inch * 2.2 inch
Anzahl Bildpunkte: 1.65 * 80 * 2.2 * 80 = 23232
a. Datenmenge = 23232 * 1 Bit =  2.90 KByte
b. Datenmenge = 23232 * 4 Bit =  11.62 KByte
c. Datenmenge = 23232 * 24 Bit =  69.70 KByte
```

16. (3 Punkte) Die Bilddaten eines Schwarzweiß-Bildes sind

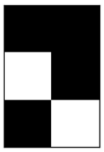
```
000000000011111110000111000111111111000011
```

a. Codiere die Bilddaten mit der Lauflängencodierung.
b. Berechne die Datenmenge der ursprünglichen Bilddaten und der komprimierten.
c. Berechne die Ersparnis durch die Kompression

Lösung:

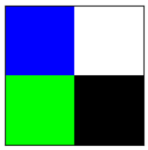
```
a. 10 7 4 3 3 9 4 2
b. Größte Zahl 10 < 16, also 4 Bit reichen aus.
c. Datenmenge ursprünglich: 42 Bit, komprimiert 8 * 4 = 32 Bit
Ersparnis: 10 Bit = 23.8%
```

17. (2 Punkte) a. Gib den Inhalt einer pbm-Datei an, die ein Bild wie unten erzeugt. Ein quadratisches Kästchen entspricht einem Bildpunkt.
b. Was bedeutet in diesem Zusammenhang die Abkürzung pbm?



Lösung: a. P1 2 3 110110
b. Portable Bitmap

18. (2 Punkte) a. Gib den Inhalt einer ppm-Datei an, die ein Farbbild wie unten erzeugt. Ein Kästchen entspricht einem Bildpunkt. Die Farben sind nicht gedruckt, es sind die Farben blau, weiß (1.Zeile) grün, schwarz (2. Zeile).
b. Was bedeutet in diesem Zusammenhang die Abkürzung ppm?



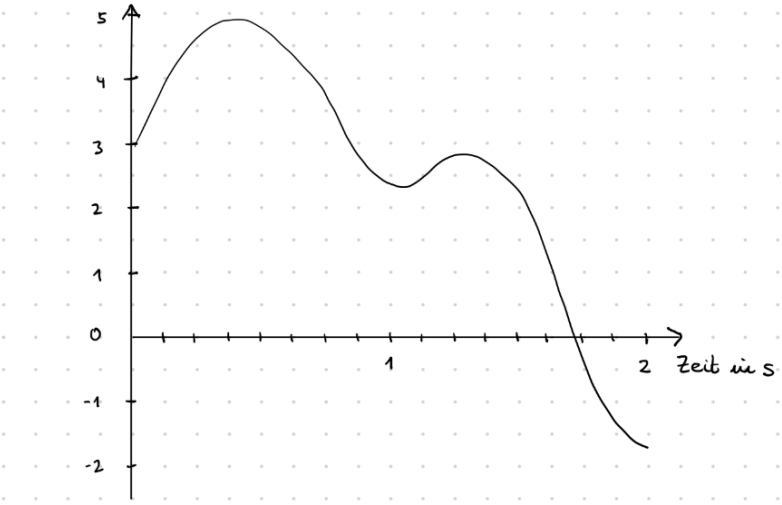
Lösung: a. P3 2 2 255 0 0 255 255 255 255 0 255 0 0 0 0
b. Portable Pixelmap

19. (2 Punkte) In einem Tonstudio werden Aufnahmen mit 24 Bit / 22050 Hz gemacht.

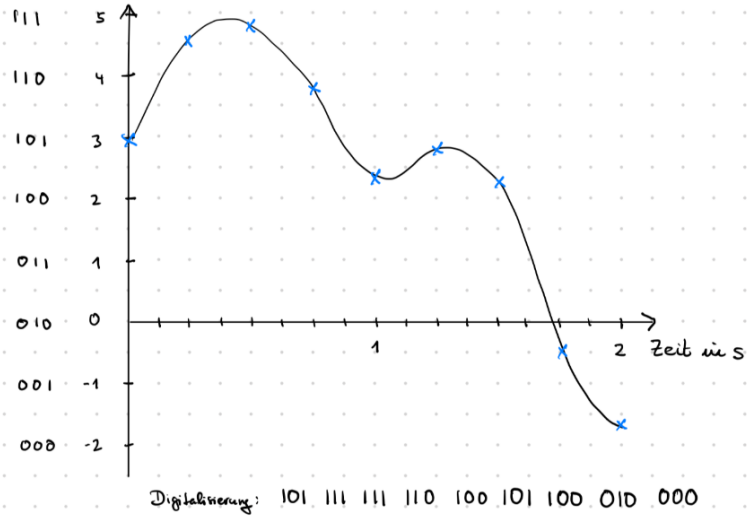
a. Berechne die Datenrate.
b. Berechne die Datenmenge in MByte für eine Aufnahme, die 1 Stunde dauert.

Lösung: a. Die Datenrate ist 529200 Bit/s
b. Die Datenmenge ist 238.14 MByte

20. (2 Punkte) Digitalisiere das abgebildete Signal mit den Parametern: Samplingrate: 4 Hz, Samplingtiefe: 3 Bit.



Lösung:



Aufgabe:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Summe:
Punkte:	2	3	3	3	3	4	3	1	2	4	1	2	1	2	3	3	2	2	2	2	48