

# **Programmering og problemløsning 5100-B1-2E16: Ugeseddel #1**

Due on Wednesday, September 14, 2016

**Mehrdad Khodaverdi [ctm546@alumni.ku.dk](mailto:ctm546@alumni.ku.dk)**

September 22, 2016

**2i.0**

Betragt EBNF'en:

```
charLiteral = ?any unicode codepoint?;
stringLiteral = '"', charLiteral, '"';
operator = '+';
expression = stringLiteral | stringLiteral, operator, expression;
```

1. Opskriv 3 forskellige gyldige expressions udelukkende ved brug af tokenerne expressions, operator og stringLiteral:

expression = stringLiteral (1)

expression = stringLiteral, operator, stringLiteral (2)

expression = stringLiteral, operator, expression (3)

2. Giv derefter eksempler på tilsvarende sekvenser, hvor tokenerne er erstattet med terminaler.

For charLiteral = a

'"a"' = '"a"' (1)

'"a"+"a"' = '"a"' '+' '"a"' (2)

'"a"+"a"+"a"' = '"a"' '+' '"a"+"a"' (3)

3. Giv et eksempel på en sekvens, som ikke er gyldig i ovenstående EBNF.

For charLiteral = a

expression  $\neq$  operator

expression  $\neq$  '+'

expression  $\neq$  stringLiteral, operator, operator

expression  $\neq$  '"a"' '+' '+'

expression  $\neq$  stringLiteral, stringLiteral, expression

expression  $\neq$  '"a"' '"a"' '"a"+"a"'

**2i.1**

Udfyld følgende tabel

Decimal	Binær	Heximal	Oktal
10	1010	A	12
21	10101	15	25
63	111111	3F	77
63	111111	3F	77

**Første række:**Decimal 10 til binær:

$$\begin{aligned}\frac{10}{2} &= 5 \quad \text{Rest } 0 \\ \frac{5}{2} &= 2 \quad \text{Rest } 1 \\ \frac{2}{2} &= 1 \quad \text{Rest } 0 \\ \frac{1}{2} &= 1 \quad \text{Rest } 1\end{aligned}$$

$$\text{Ergo } (10)_{10} = (1010)_2$$

Det er givet at

Dec	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

$$\text{Derfor må } (10)_{10} = (A)_{Hex}$$

Decimal 10 til Oktal:

$$10 \bmod 8 = 2 \quad \text{hvilket man regner ved: } \frac{10}{8} = \underline{1}.25 \Rightarrow 8^1 + 2 = 10 \quad \text{altså rest } 2$$

Vores Octal nummer er nu ??2 og vi gentager igen fra før:

$$1 \bmod 8 = 1 \quad \text{hvilket man regner ved: } \frac{1}{8} = \underline{0}.125 \Rightarrow 8^0 + 1 = 1 \quad \text{altså rest } 1$$

Ergo tallet er 12 .

$$\text{Derfor må } (10)_{10} = (12)_8$$

**Anden række:**Binær 10101 til decimal:

$$\begin{array}{ccccc} 1 & 0 & 1 & 0 & 1 \\ 2^4 & +2^3 & +2^2 & +2^1 & +2^0 \end{array}$$

$$10101 \text{ betyder altså } 2^0 + 2^2 + 2^4 = 21$$

$$\text{Ergo } (10101)_2 = (21)_{10}$$

**10101 til Hex:**

Vi udvider vores tabel gennem samme øvelse som vist oven over.:

Dec	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bin	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Vi ser igen på 10101. Da vi ved at 1 Hex ækvivalent med 4 bit deler vi det binær tal op i grupper af 4. 10101 skrives som 0001 0101 og fra tabellen har vi:

$$\begin{array}{cc} 0001 & 0101 \\ 1 & 5 \end{array}$$

$$\text{Ergo } (10101)_2 = (15)_{Hex}$$

10101 til Oktal:

Vi udvider vores tabel nu med oktal:

Table 1: Konverterings tabel

Dec	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bin	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Oktal	0	1	2	3	4	5	6	7								

Vi ser på 10101. Da vi ved at 1 oktal er ækvivalent med 3 bit deler vi det binær tal op i grupper af 3. 10101 skrives som 010 101 og ved brug af tabellen får:

$$\begin{array}{cc} 010 & 101 \\ 2 & 5 \end{array}$$

$$\text{Ergo } (10101)_2 = (25)_8$$

**Tredje række:**

Hex 3F til Bin og til oktal:

Omskriver hex 3F til bin:  $3 \rightarrow 0011$  og  $F \rightarrow 1111$ . Ergo  $(3F)_{Hex} = (00111111)_2$

Nu deler vi det binær tal op i grupper af 3 og får 000 111 111. Fra tabellen har vi:

$$\begin{array}{ccc} 000 & 111 & 111 \\ 0 & 7 & 7 \end{array}$$

$$\text{Ergo } (3F)_{Hex} = (77)_8$$

Hex til decimal:

$$3F = (3 * 16^1) + (15 * 16^0) = 48 + 15 = 63$$

$$\text{Ergo } (3F)_{Hex} = (63)_{10}$$

**Fjerde række:**

oktal 77 til bin også til hex

Vi ser hurtigt fra tabel 1 at tallet 77 kan skrives som binært:

$$\begin{array}{cc} 7 & 7 \\ 111 & 111 \end{array}$$

$$\text{Ergo } (77)_8 = (111111)_2$$

og igen kan det grupperes som 0011 1111 hvilket giver henholdsvis tallet 3 og F, ergo 3F

**2i.2**

Først defineres **a** som "Hello World"

Derefter printer jeg de første fire char i a, for til sidst at printe fra den sjette char til slut.

```
> let a = "Hello World";;  
  
val a : string = "Hello World"  
  
> a[..4];;  
val it : string = "Hello"  
> a[6..];;  
val it : string = "World"  
  
> a[6..]+", "+a[..4]+"!";;  
val it : string = "World, Hello!"
```