Try-out 7: Stringbewerking

Plaats een string in het geheugen in kleine letters. Lees deze string uit en toon hem op het scherm met kleine letters. Aangezien je nog geen voorwaardelijke structuren hebt gezien in assembler, mag je de stringlengte vast coderen.
Om een geheugenadres uit te lezen gebruik je de volgende syntax:

mov reg,[adres] of mov reg,[reg]
Gezien je hier met stringbewerkingen bezig bent, is het meest correct als je ook gebruik maakt van de indexregisters (SI en/of DI).
Als je bijvoorbeeld “appelgebak” in geeft, zet je de teller op 10 en genereer je de volgende uitvoer.

```
C:\> TRY7
APPELGEBAK
```

Try-out 7: Oplossing

```
e 120 "appelgebak"
A
MOV     CX,000A     ; Plaats 10 in het countregister
MOV     SI,120      ; wijs met het SI-register naar de string
MOV     AH,02       ; subfunctie 02
MOV     DL,[SI]     ; haal een karakter uit de string op op het adres opgegeven door SI
SUB     DL,20       ; tel 20H af om naar hoofdletter te gaan
INT     21          ; print de hoofdletter
INC     SI          ; verhoog SI met 1; is hetzelfde als ADD SI,01
LOOP    0108        ; spring naar adres 108
INT     20          ; einde
```

Try-out 8: Stringbewerking

Plaats een string in het geheugen bestaande uit zowel kleine als hoofdletters. Lees deze string uit en toon hem op het scherm enkel kleine letters. Aangezien je nog geen voorwaardelijke structuren hebt gezien in assembler, mag je de stringlengte vast coderen.
Hint: gebruik een logische operator om dit probleem op te lossen.
Als je bijvoorbeeld “RaRa welke Logische Operator” in geeft, zet je de teller op 28 en genereer je de volgende uitvoer.

Door een wijziging van de logische operator en operand kan je het programma wijzigen dat enkel hoofdletters getoond worden.

```
C:\> TRY8
rara welke logische operator
```

Try-out 8: Stringbewerking

Verklaring: "Het gebruik van de AND – OR operator"

OR / AND

A	B	A OR B	A AND B
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

Als deze instructie wordt uitgevoerd op een register;

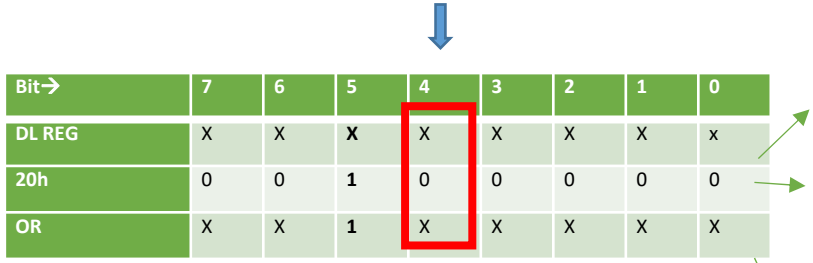
Bit→	7	6	5	4	3	2	1	0
DL REG	B7	B6	B5	B4	B3	B2	B1	B0
XX	B7'	B6'	B5'	B4'	B3'	B2'	B1'	B0'
OR	B7 OR B7'	B6 OR B6'	B5 OR B5'	B5 OR B5'	B4 OR B4'	B3 OR B3'	B2 OR B2'	B1 OR B1'

Try-out 8: Stringbewerking

Verklaring: "Het gebruik van de AND – OR operator"

In Try-out 8 wordt gevraagd om een karakter (hoofdletter of kleine letter), om te zetten naar een kleine letter.
→ Dit kan door gebruik te maken van de OR functie.

- In de ASCII tabel zie je, dat voor al de kleine letters, de 5de bit 1 is !
- Door het commando "OR DL, 20", zal het DL register worden overgenomen, maar de 5de bit op één!



Bit→	7	6	5	4	3	2	1	0
DL REG	X	X	**X**	X	X	X	X	x
20h	0	0	**1**	0	0	0	0	0
OR	X	X	**1**	X	X	X	X	X

OR DL, 20

Karakter in het DL register

De OR instructie wordt uitgevoerd met 20(h), hierdoor staat op de 5de bit steeds een één. (20h = 0010 0000)

Na de uitvoering van de OR instructie, zal de 5de bit steeds een 1 zijn.
(X or 1 = 1) Ongeacht wat de oorspronkelijke waarde van het DL register is.
De andere bits worden overgenomen van het DL. (X OR 0 = X)
Als de 5de bit 1 is, zal het karakter altijd getoond worden als een kleine letter.

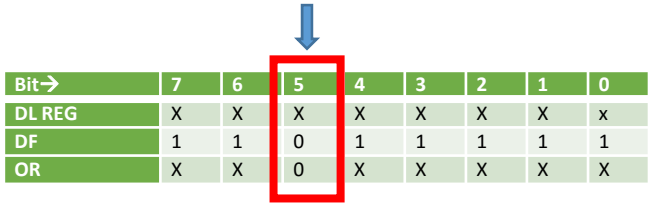
Try-out 8: Stringbewerking

Verklaring: "Het gebruik van de AND – OR operator"

Een karakter (hoofdletter of kleine letter), om te zetten naar een hoofdletter.
→ Dit kan door gebruik te maken van de AND functie.

- In de ASCII tabel zie je dat voor AL de kleine letters, de 5de bit 0 is !
- Door het commando "AND DL, DF", zal het DL register worden overgenomen, met de 5de bit op nul.

AND DL, DF



Bit→	7	6	5	4	3	2	1	0
DL REG	X	X	X	X	X	X	X	x
DF	1	1	0	1	1	1	1	1
OR	X	X	0	X	X	X	X	X

Zoek welk commando je kan gebruiken om voor de getallen 0-9 in de ASCII tabel te bepalen als het opgegeven getal oneven is. (tip: oneven getal → b0 is altijd 1)

Try-out 8: Stringbewerking

```
e 120 "ApPeLgEbAk"
A
MOV     CX,000A     → plaats 10 in het countregister
MOV     SI,120      → wijs met het SI-register naar de string
MOV     AH,02       → subfunctie 02
MOV     DL,[SI]     → haal een karakter uit de string op op het adres opgegeven door SI
OR      DL,20       → met OR 20 zal bitpositie 5 altijd op 1 staan zodat het altijd een kleine letter is! *
INT     21          → print de hoofdletter
INC     SI          → verhoog SI met 1; is hetzelfde als ADD SI,01
LOOP    0108        → spring naar adres 108
INT     20          → einde
```

* Verander deze regel door “AND DL, DF” voor hoofdletters