

Assembler, de eerste stappen ...

Oefeningen

1. Print één karakter (met INT 21, sub 02)

a

2. Print in een loop 10x één karakter (met INT 21, sub 02 + LOOP)

aaaaaaaaaa

3. Print drie karakters (INT 21, SUB 02)

abc

4. Print in een loop 5x drie karakter gescheiden met een spatie (INT21, SUB 02 + LOOP)

abc abc abc abc abc

5. Print een reeks van karakters (INT 21, SUB 09)

abcdefghijklmnopqrstuvwxyz

6. Print een reeks van karakters 5x na elkaar (INT 21, SUB 09, LOOP)

test test test test test

7. Vraag een karakter (INT 21 SUB 01) en print dit karakter 5x af (INT 21 + SUB02 + LOOP)

tttttt

Assembler, de eerste stappen ...

Oefeningen

1. Print één karakter (met INT 21, sub 02)

a

```
-a
1469:0100 MOV AH, 02
1469:0102 MOV DL, 61
1469:0104 INT 21
1469:0106 INT 20
1469:0108
-g
a
Program terminated normally
-
```

Minimale CX waarde t.b.v opslag = 08

Assembler, de eerste stappen ...

Oefeningen

2. Print in een loop 10x één karakter (met INT 21, sub 02 + LOOP)

aaaaaaaaaa

```
-a
1469:0100 MOV CX, 000A
1469:0103 MOV AH, 02
1469:0105 MOV DL, 61
1469:0107 INT 21
1469:0109 LOOP 107
1469:010B INT 20
1469:010D
-g
aaaaaaaaaa
Program terminated normally
-
```

Minimale CX waarde t.b.v opslag = 0D

Assembler, de eerste stappen ...

Oefeningen

3. Print drie karakters (INT 21, SUB 02)

abc

```
-a
1469:0100 MOV AH, 02
1469:0102 MOV DL, 61
1469:0104 INT 21
1469:0106 MOV DL, 62
1469:0108 INT 21
1469:010A MOV DL, 63
1469:010C INT 21
1469:010E INT 20
1469:0110
-g
abc
Program terminated normally
-
```

Minimale CX waarde t.b.v opslag = 10

Assembler, de eerste stappen ...

Oefeningen

4. Print in een loop 5x drie karakter gescheiden met een spatie (INT21, SUB 02 + LOOP)

abc abc abc abc abc

```
-a
1469:0100 MOV CX, 0005
1469:0103 MOV AH, 02
1469:0105 MOV DL, 61
1469:0107 INT 21
1469:0109 MOV DL, 62
1469:010B INT 21
1469:010D MOV DL, 63
1469:010F INT 21
1469:0111 MOV DL, 20
1469:0113 INT 21
1469:0115 LOOP 105
1469:0117 INT 20
1469:0119
-g
abc abc abc abc abc
Program terminated normally
-
```

Minimale CX waarde t.b.v opslag = 19

Assembler, de eerste stappen ...

Oefeningen

5. Print een reeks van karakters (INT 21, SUB 09)

abcdefghijklmnopqrstuvwxyz

```
-a 100
1469:0100 MOV AH, 09
1469:0102 MOV DX, 120
1469:0105 INT 21
1469:0107 INT 20
1469:0109
-e 120 "abcdefghijklmnopqrstuvwxyz" 24
-g
abcdefghijklmnopqrstuvwxyz
Program terminated normally
```

OPTIE 01= Het alfabet als reeks van karakters (subfunctie 09) .

Minimale CX waarde t.b.v opslag = 3C

```
-a
1469:0100 MOV CX, 001A
1469:0103 MOV AH, 02
1469:0105 MOV DL, 61
1469:0107 INT 21
1469:0109 INC DL
1469:010B LOOP 107
1469:010D INT 20
1469:010F
-g
abcdefghijklmnopqrstuvwxyz
Program terminated normally
```

OPTIE 02= Het alfabet een herhaling van karakters (Sub 02 + loop).

Minimale CX waarde t.b.v opslag = F

Assembler, de eerste stappen ...

Oefeningen

6. Print een reeks van karakters 5x na elkaar (INT 21, SUB 09, LOOP)

test test test test test

```
-a 100
1469:0100 MOV CX, 0005
1469:0103 MOV AH, 09
1469:0105 MOV DX, 120
1469:0108 INT 21
1469:010A LOOP 108
1469:010C INT 20
1469:010E
-e 120 "test $"
-g
test test test test test
Program terminated normally
```

Minimale CX waarde t.b.v opslag = 27

Assembler, de eerste stappen ...

Oefeningen

7. Vraag een karakter (INT 21 SUB 01) en print dit karakter 5x af (INT 21 + SUB02 + LOOP)

tttttt

```
-a
1469:0100 MOV AH, 01
1469:0102 INT 21
1469:0104 MOV DL, AL
1469:0106 MOV CX, 0005
1469:0109 MOV AH, 02
1469:010B INT 21
1469:010D LOOP 010B
1469:010F INT 20
1469:0111
-g
tttttt
Program terminated normally
-
```

Minimale CX waarde t.b.v opslag = 11