

◆ Try-outs in Assembler

Try-out 1: INT 21

Maak twee verschillende programma's in assembler.

Doel van het programma is je voornaam 10 maal achter elkaar printen gescheiden door spaties. Beide programma's moeten hetzelfde resultaat geven.

Met verschillende programma's wordt bedoeld dat er gebruik wordt gemaakt van verschillende DOS-subfuncties.

- Voor TRY1a.COM maak je gebruik van subfunctie 02 van INT21 (printen van een teken).
- Voor TRY1b.COM maak je gebruik van subfunctie 09 van INT21 (printen van een string).

```
C:\>try1a
JOHAN JOHAN JOHAN JOHAN JOHAN JOHAN JOHAN JOHAN JOHAN JOHAN
C:\>
C:\>try1b
JOHAN JOHAN JOHAN JOHAN JOHAN JOHAN JOHAN JOHAN JOHAN JOHAN
C:\>
```

Maak de programma's zo optimaal mogelijk.

Try-out 2: Regel vullen

Schrijf een gestructureerd programma in assembler (TRY2.COM), waarbij een volledige

regel (80 karakters) wordt opgevuld met het karakter met ASCII-waarde 2 (het teken ☺).

[illegible]

Pas dit programma aan zodat een volledig scherm opgevuld wordt (25 lijnen)

Try-out 3: Omzetting van "case"

Het programma TRY3.COM laat toe om een letter in te geven (subfunctie 01 van INT 21). Er mag verondersteld worden dat er een hoofdletter wordt ingegeven (controleren is hier dus nog niet nodig).

Deze hoofdletter wordt vervolgens omgezet naar een kleine letter en op de volgende positie op het scherm geplaatst.

```
C:\>TRY3
Gg
```

Try-out 4: Letter ingeven en afprinten : oefening in debuggen

Maak een assemblerprogramma in debug waardoor het mogelijk wordt een letter in te geven. Deze letter dient dan tien keer op het scherm te worden gedrukt, zoals hieronder wordt aangegeven (**NIET** gescheiden door een spatie).

```
C:\>TRY4-1
SSSSSSSSSS
C:\>
```

Pas dit programma aan door de tekens te scheiden door middel van een linefeed of een spatie. Je zal merken dat dit programma niet de verwachte output geeft. Probeer de fout te vinden door te debuggen (stap voor stap de werking van elke instructie in je programma doorlopen en controleren op zijn juiste werking d.m.v. trace en proceed). Als je de fout gevonden heb, probeer ze dan op te lossen.

```
C:\>TRY4-2
S
 S
  S
   S
    S
     S
      S
       S
        S
C:\>
```

Try-out 5: Alfabet in hoofd- en kleine letters

Maak een assemblerprogramma in debug dat het volledig alfabet toont, afwisselend met hoofdletters en kleine letters.

```
C:\>TRY5
AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz.
C:\>
```

Try-out 6: Piece of cake

Schrijf een programma waarmee vanuit het geheugen de string "Piece of cake" tien keer op het scherm wordt geplaatst.

```
C:\>TRY6
Piece of cake
Piece of cake
Piece of cake
Piece of cake
Piece of cake
Piece of cake
Piece of cake
Piece of cake
Piece of cake
Piece of cake
C:\>
```

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 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
C:\>
```

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.

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

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

Try-out 9: INT10

Schrijf een programma waarmee vanuit het geheugen je voornaam in kleur (vrij te kiezen) 1 keer op het scherm wordt geplaatst op rij 12, kolom 40. Maak gebruik van interrupt 10 (subfunctie 00 en 13)

```
C:\>TRY9
```

Johan

```
C:\>
```

❖ Oefeningen op push en pop

Try-out 10: Compu Compu Compu Compu Compu Compu

Schrijf een programma waarmee we de string "compu" acht keer onder elkaar zetten, met telkens een andere achtergrondkleur.

Eerste positie: rij = 8, kolom = 25

Try-out 11: Nog kleurigere Compu

Pas het vorige programma zo aan dat we de string "compu" 16 keer onder elkaar zetten, maar telkens met een andere voorgrondgrondkleur.

Eerste positie: rij = 01; kolom = 25

Try-out 12: Scherm vol krijgen

Schrijf een programma waarmee we de tien eerste regels van het scherm voldrukken met het karakter met ASCII-waarde 2 (teken ☺).

Doe dit met int 10 en AH=13.

De string in het geheugen bevat dus enkel de waarde 2.

Try-out 13: Pizzaland

Schrijf een programma waarmee we de Italiaanse vlag (Belgisch mag ook natuurlijk) op een grafisch scherm tonen. Grafische scherminstelling via : int 10 AH=00 en AL=12.

Een grafische pixel plaatsen op een grafisch scherm kan met INT 10 met subfunctie "AH=0Ch"

Tip: Gebruik het CX register zowel voor de kolombepaling als voor de teller (op deze manier vermijd je overbodige push & pop operaties).

Try-out 14: Broodje Krokot

Schrijf een programma waarmee we de Nederlandse vlag (bij ernstige morele bezwaren mag je ook een andere vlag met horizontale strepen kiezen) op een grafisch scherm tonen. Grafische scherminstelling via: int 10 AH=00 en AL=12.

◆ Oefeningen op compare en jump**Try-out 15: Hoofdletter of niet**

Schrijf een programma waarmee een van ingegeven karakter (subfunctie 01 van INT 21) gecontroleerd wordt of het een hoofdletter is of niet? Geef de meldingen in de aard van "Hoera, 't is er een" en "pech, 't is er geen" naargelang het resultaat.

Try-out 16: Letters omzetten

Schrijf een programma waarmee een van ingegeven karakter (subfunctie 01 van INT 21) gecontroleerd wordt of het een hoofdletter is of niet en waarna deze wordt omgezet van case. Als het geen letter is wordt ze gewoon zonder meer terug afgedrukt.