

2021 januari examen

Oplossingen voor de oefeningen van het januari examen met dank aan [ISW](#) en Jens Gervais:

Oefening 1:

```

Name
Size
Modified
1_ComputersystemenAssemblyOefening1_JensGervais
.txt
Actions
1 KB5 months ago
1_ComputersystemenAssemblyOefening1Invoer_JensGervais
.txt
Actions
< 1 KB5 months ago
1_ComputersystemenAssemblyOefening2_JensGervais
.txt
Actions
< 1 KB5 months ago
3 files2 KB
Nextcloud - ISW Leuven
Privacy policy

Get your own free account

%include "gt.asm"
covar
outarea: resb 70
    DB 0Dh, 0Ah
inarea: resb 70
```

```
□aantalkm: resd 1
□liters: resd 1
□duizend: dd 1000
□verbruik: resd 1
□verbruikLagerDan: resd 1
□zestig: dd 60
□totaal: resd 1
□honderd: dd 100
□tekst: DB '% van de wagens heeft een verbruik onder de 60.'
```

inleiding

```
□openin ; open het invoerbestand
□openuit ; open het uitvoerbestand
□cld
```

lijnlezen:

```
□lees
□cmp eax, 0
□je einde
□mov edx, [totaal]
□add edx, 1
□mov [totaal], edx
□call lijnvullen
□mov ecx, 34
□mov esi, inarea
□mov edi, outarea
□rep movsb

□mov ecx, 5
□mov esi, inarea + 21
□tekstbin
□mov [aantalkm], eax
```

```
□mov ecx, 4
□mov esi, inarea + 31
□tekstbin
□mov [liters], eax
```

```
□mov eax, [liters]
□imul dword [duizend]
□idiv dword [aantalkm]
□mov [verbruik], eax
```

```
mov eax, [zestig]
```

```
cmp [verbruik], eax
```

```
je lijnlezen
```

```
mov edx, [verbruikLagerDan]
```

```
add edx, 1
```

```
mov [verbruikLagerDan], edx
```

```
mov ecx, 2
```

```
mov edi, outarea + 42
```

```
mov eax, [verbruik]
```

```
call asciiBerekening
```

```
schrijf
```

```
jmp lijnlezen
```

```
einde:
```

```
call lijnvullen
```

```
schrijf
```

```
mov eax, [verbruikLagerDan]
```

```
imul dword [honderd]
```

```
idiv dword [totaal]
```

```
mov edi, outarea + 3
```

```
call asciiBerekening
```

```
mov esi, tekst
```

```
mov edi, outarea + 4
```

```
mov ecx, 47
```

```
rep movsb
```

```
schrijf
```

```
slot
```

```
lijnvullen:
```

```
mov ecx, 70
```

```
mov al, ' '
```

```
mov edi, outarea
```

```
rep stosb
```

```
sub edi, edi
```

```
ret
```

asciiBerekening:

```
mov ebx, 10
```

```
std
```

```
lus:
```

```
mov edx, 0
```

```
idiv ebx
```

```
or dl, 30h
```

```
xchg al, dl
```

```
stosb
```

```
xchg al, dl
```

```
cmp eax, 0
```

```
jne lus
```

```
cld
```

```
ret
```

Oefening 1 invoer:

Name

Size

Modified

1_ComputersystemenAssemblyOefening1_JensGervais

.txt

Actions

1 KB 5 months ago

1_ComputersystemenAssemblyOefening1Invoer_JensGervais

.txt

Actions

< 1 KB 5 months ago

1_ComputersystemenAssemblyOefening2_JensGervais

.txt

Actions

< 1 KB 5 months ago

3 files 2 KB

Nextcloud - ISW Leuven

[Privacy policy](#)

[Get your own free account](#)

OPEL MERIVA	855	45
OPEL ASTRA	935	47
FORD Focus	1230	62
BMW 735i	1100	85
VW GOLF GTD	895	44
FIAT CROMA 1.9	725	39
Tesla Model S	5	0

Oefening 2:

□

Name

Size

Modified

□

1_ComputersystemenAssemblyOefening1_JensGervais

.txt

Actions

1 KB□5 months ago

□

1_ComputersystemenAssemblyOefening1Invoer_JensGervais

.txt

Actions

< 1 KB□5 months ago

□

1_ComputersystemenAssemblyOefening2_JensGervais

.txt

Actions

< 1 KB□5 months ago

3 files□2 KB□

Nextcloud - ISW Leuven

[Privacy policy](#)

[Get your own free account](#)

Bereken fibonacci (https://en.wikipedia.org/wiki/Fibonacci_number)

```
%include "gt.asm"

cvar
[x: resd 1
[eersteFibo: dd 0
[tweedeFibo: dd 1
[nieuweFibo: resd 1

inleiding
[inv [x]
[uit [eersteFibo]
[uit [tweedeFibo]
[mov ecx, [x]
[sub ecx, 2

vlgFibo:
[mov eax, [eersteFibo]
[add eax, [tweedeFibo]
[mov [nieuweFibo], eax
[uit [nieuweFibo]
[mov eax, [tweedeFibo]
[mov [eersteFibo], eax
[mov eax, [nieuweFibo]
[mov [tweedeFibo], eax
[loop vlgFibo

slot
```

Revision #1

Created 17 June 2021 13:30:42 by Jasper G.

Updated 3 December 2021 22:13:09 by Jasper G.