TietgenSkolen
EDB-Skolen

Datamatiker uddannelsen

Opgaver til faget

Maskinarkitektur & Operativsystemer

Udgave 1.0

Samlet og redigeret af
Bjørk Busch

Forord.

Opgavesamlingen indeholder en række opgaver vedrørende forskellige emner til faget "maskinarkitektur & operativsystemer".

Indholdsfortegnelse:

1.

Datarepræsentation

Opgaver

Indhold:

1.1.a

Gennerel Tallære og datarepræsentation OPGAVER

Bjørk Busch

1) Omregn følgende binære tal (uden fortegn) til decimalt:

a)	1101	b)	1010101
c)	11000111	d)	10010101111
e)	101,101	f)	10010,0011

2) Omregn følgende decimale tal til binære:

a)	27	b)	216
c)	312	d)	5312
e)	5,75	f)	10,3

3) Omregn følgende hexadecimale tal (uden fortegn) til decimale:

a)	13	b)	2AF
c)	F13	d)	F132

4) Omregn følgende decimale tal til hexadecimale:

a)	21	b)	271
c)	5312	d)	12121

5) Omregn følgende binære tal (uden fortegn) til oktale og hexadecimale uden brug af det decimale talsystem:

a)	01011101	b)	10011100
c)	11101011	d)	01100101

6) Omregn følgende oktale tal (uden fortegn) til hexadecimale uden brug af det decimale talsystem:

a)	172	b)	123
c)	312	d)	234

7) Omregn følgende hexadecimale tal (uden fortegn) til oktale uden brug af det decimale talsystem:

a)	0F1	b)	1CA
c)	312	d)	2EB

8) Vis hvorledes følgende decimale tal kan repræsenteres binært på en 8-bit maskine:

a)	4	b)	-4
c)	112	d)	-112
e)	-1	f)	255

9) Vis hvorledes, tallene fra foregående opave vil blive udskrevet ved et hexadecimalt dump og ved et oktalt dump.

10) Ved hex-dump af det interne lager af en tekst i ASCII format fremkommer følgede:

54 65 6B 73 74 20 69 20 41 53 43 49 49

Oversæt dumpet til klar tekst.

11) Ved hex-dump af det interne lager af et talfelt fremkommer følgede:

34 38

Hvad er det for et tal decimalt når:

a) Tallet er repræcenteret binært.
b) Tallet er repræcenteret i ascii-kode.

12) På en ordmaskine med 36-bits ord er der fremkommet et oktal-dump af et ord med følgende indhold:

073 730 307 411

Ordet er opdelt i kvartord, der hver indeholder et heltal med fortegn.
Vis de fire tal decimalt.
Vis hvorledes man kan omregne fra oktalt til hex uden brug af det decimale talsystem, illustreret ved de ovennævnte fire tal.

13) Vis hvorledes følgende tal kan repræsenteres som binær integer (16-bit) og i ascii-format (begge udskrevet i hex-dump-format):

a) 123 b) -123

14) Vis indholdet af integer-variablerne (16 bit) binært og hexadecimalt (uden ombytning) efter udførelsen af følgende pascal-instruktioner:

```
F1 := 743;   F2 := -313;   F3 := -200;
```

15) Ved dump af integer-felter i et pascal-program har felterne følgende indhold (skal ikke ombyttes):

F1: FF27 F2: 0564 F3: 8234

Vis indholdet decimalt.

16) I et pascalprogram findes følgende instruktion:

C := A + B;

Alle felter er af typen integer (16 bit).
Vis hvorledes udregningen foretages BINÆRT når:

a) A = 515 og B = 519 (decimalt)
b) A = -515 og B = 65 (decimalt)

17) Et andet sted i pascalprogramet findes følgende instruktion:

C := A - B;

Alle felter er af typen integer.
Vis hvorledes udregningen foretages BINÆRT på en maskine uden subtractor (der anvendes kun negation og addition) når:

a) A = 515 og B = 519 (decimalt)
b) A = 515 og B = 65 (decimalt)
c) A = -515 og B = -65 (decimalt)
d) A = -32.000 og B = 32.000 (decimalt)

18) Vis indholdet af integer-variablerne (16 bit) binært og hexadecimalt (uden ombytning) efter udførelsen af følgende pascal-instruktioner:

F1 := 10; F2 := 317; F3 := -23;

19) Ved dump har nogle integer-variabler fra et pascal-program følgende indhold (skal ikke ombyttes):

a) FF12 b) 0234 c) 1234 d) 8001

Vis indholdet decimalt

20) I et pascalprogram findes følgende instruktion:

C := A - B;

Vis hvorledes udregningen foretages binært ved brug af negation og addition, idet variablerne alle er af typen integer (16 bit) og har følgende indhold:

A = 120 og B = 47 (decimalt)

21) Omregn følgende binære tal (uden fortegn) til decimalt:

a)	11101	b)	110101
c)	101000101	d)	111001011001
e)	1100,11101	f)	10,011011

22) Omregn følgende decimale tal til binære:

a)	1027	b)	51202
c)	678	d)	1412
e)	0,78	f)	14,13

23) Omregn følgende hexadecimale tal (uden fortegn) til decimale:

a)	113	b)	BEA
c)	E04	d)	FABE

24) Vis hvorledes følgende decimale tal kan repræsenteres binært på en 8-bit maskine:

a)	14	b)	-14
c)	101	d)	-101
e)	-56	f)	250

25) Vis hvorledes, tallene fra foregående opave vil blive udskrevet ved et hexadecimalt dump og ved et oktalt dump.

26) På en ordmaskine med 36-bits ord er der fremkommet et oktal-dump af et ord med følgende indhold:

031 424 313 717

Ordet er opdelt i kvartord, der hver indeholder et heltal med fortegn.
Vis de fire tal decimalt.

Vis hvorledes man kan omregne fra oktalt til hex uden brug af det decimale talsystem, illustreret ved de ovennævnte fire tal.

27) Vis hvorledes følgende tal kan repræsenteres som binær integer (16-bit) og i ascii-format (begge udskrevet i hex-dump-format):

a)	432	b)	-432

28) Omregn følgende tal til de andre talsystemer.
Tallene er på 16-bits med fortegn (2-komplement).

	Decimal	Binær	Octal	Hex
a)			003361	
b)	549			
c)		1111110110011001		
d)			005221	
e)		0000010101000101		
f)				F910
g)			005662	
h)				0FB9
i)			010467	
j)	1153			
k)		1110111110001111		
l)				EED9

1.1.b

Gennerel Tallære og datarepræsentation
LØSNINGER

Bjørk Busch

Løsning 1

a) Multiplikationsmetoden.

```
 1            1                 0                 1
 1   (1*2) + 1 = 3   (3*2) + 0 = 6   (6*2) + 1 = 13
```

b) Potensmetoden.

```
1       0       1       0       1       0       1
1*2^6 + 0*2^5 + 1*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 1*2^0 =
64    + 0     + 16    + 0     + 4     + 0     + 1     = 85
```

c) Opslagsmetoden for max 8 Bit.

$1100\ 0111_2 = C7_{16} =$ **199**

d) Tabelmetoden.

```
1       0 0 1     0 1    0 1   1   1   1
1024 +     128 +   32 +   8 + 4 + 2 + 1   =   1199
```

e) 101,101 (2) = 4 + 1 + 1/2 + 1/8 = **5 5/8**

f) 10010,0011 (2) = 16 + 2 + 1/8 + 1/16 = **18 3/16**

Løsning 2

a) Divisionsmetoden.

```
  27        ulige                          1
  13        ulige                    1
   6        lige                     0
   3        ulige        1
   1        ulige 1
   0        slut
-----------------------------------------
                        1     1     0     1     1
```

b) Opslagsmetoden.

216 = D8 (16) = **1101 1000 (2)**

c) Potens-/subtraktionsmetoden.

```
  312 =    256  + 56                256 = 2^8
   56 =     32   + 24                32 = 2^5
   24 =     16   +  8                16 = 2^4
    8 =      8   +  0                 8 = 2^3
312  =  1 0011 1000 (2)
```

d) 5312 = **1 0100 1100 0000 (2)**

e) 5,75 = **101,11 (2)**

f) 10,3 = **1010,0101 (2)** med 4 dec. nøjagtighed

Løsning 3

a) 13 (16) = 16 + 3 = **19**

b) 2AF (16) = 2*256 + 10*16 + 15 = **687**

c) F13 (16) = 15*256 + 16 + 3 = **3859**

d) F132 (16) = 15*4096 + 256 + 3*16 + 2 = **61746**

Løsning 4

a) 21 = **15 (16)**
b) 271 = **10F (16)**
c) 5312 = **14C0 (16)**
d) 12121 = **2F59 (16)**

Løsning 5

a) 01011101 (2) = 01 011 101 (2) = **135 (8)**
= 0101 1101 (2) = **5D (16)**

b) 10011100 (2) = **234 (8)** = **9C (16)**
c) 11101011 (2) = **353 (8)** = **EB (16)**
d) 01100101 (2) = **145 (8)** = **65 (16)**

Løsning 6

a) 172 (8) = 01 111 010 (2) = 0111 1010 (2) = **7A (16)**
b) 123 (8) = 01 010 011 (2) = 0101 0011 (2) = **53 (16)**
c) 312 (8) = 11 001 010 (2) = 1100 1010 (2) = **CA (16)**
d) 234 (8) = 10 011 100 (2) = 1001 1100 (2) = **9C (16)**

Løsning 7

a) 0F1 (16) = 0000 1111 0001 (2) =
000 011 110 001 (2) = **0361 (8)**

b) 1CA (16) = 0001 1100 1010 (2) =
000 111 001 010 (2) = **0712 (8)**

c) 312 (16) = 0011 0001 0010 (2) =
001 100 010 010 (2) = **1422 (8)**

d) 2EB (16) = 0010 1110 1011 (2) =
001 011 101 011 (2) = **1353 (8)**

Løsning 8

a) 4 = **0000 0100 (2)**

b) 4 = 0000 0100 (2)

1111 1011 (2) (1-komplement)
\+ 1 (2)

-4 = **1111 1100 (2)**

c) 112 = **0111 0000 (2)**

d) 112 = 0111 0000 (2)

1000 1111 (2) (1-komplement)
\+ 1 (2)

-112 = **1001 0000 (2)**

e) 1 = 0000 0001 (2)

1111 1110 (2) (1-komplement)
\+ 1 (2)

-1 = **1111 1111 (2)**

f) 255 = **1111 1111 (2)** fortegnsløst

Løsning 9

```
a)    00 00 0 100 (2)  =  04 (16)  =  004 (8)
b)    11 11 1 100 (2)  =  FC (16)  =  374 (8)
c)    01 11 0 000 (2)  =  70 (16)  =  160 (8)
d)    10 01 0 000 (2)  =  90 (16)  =  220 (8)
e)    11 11 1 111 (2)  =  FF (16)  =  377 (8)
f)    11 11 1 111 (2)  =  FF (16)  =  377 (8)
```

Løsning 10

```
54 65 6B 73 74 20 69 20 41 53 43 49 49
T  e  k  s  t     i     A  S  C  I  I
```

Løsning 11

```
a)    34 38 (16)  =  0011 0100 0011 1000 (2)  =  13.368
b)    34 38 (16)  =  48
```

Løsning 12

Omsætning til decimal:

```
073 (8)  =  000 111 011 (2)  =  59 (10)

730 (8)  =  111 011 000 (2)    (negativ)

            000 100 111 (2)    (1-komplement)
          +           1 (2)
          -----------------
            000 101 000 (2)  =  40 (10)

730 (8)  =  -40 (10)
307 (8)  =  011 000 111 (2)  =  199 (10)

411 (8)  =  100 001 001 (2)    (negativ)

              011 110 110 (2)  (1-komplement)
            +           1 (2)
            -----------------
              011 110 111 (2)  =  247 (10)

411 (8)  =  -247 (10)
Omsætning til hex:
073 (8) = 000 111 011 (2) = 0 0011 1011 (2) = 03B (16)
730 (8) = 111 011 000 (2) = 1 1101 1000 (2) = 1D8 (16)
307 (8) = 011 000 111 (2) = 0 1100 0111 (2) = 0C7 (16)
411 (8) = 100 001 001 (2) = 1 0000 1001 (2) = 109 (16)
```

Løsning 13

```
a)    123  =  0000 0000 0111 1011 (2)
           =  007B (16)  binært-format.
      123  =  31 32 33 (16)  ASCII-format.

b)    -123 negativ tal.

      123  =      0000 0000 0111 1011 (2)

                  1111 1111 1000 0100 (2) (1-komplement)
            +                       1 (2)
            -------------------------
      -123 =      1111 1111 1000 0101 (2)  (2-komplement)
           =   FF 85 (16)  hex-format.
      -123 =  2D 31 32 33 (16)  ASCII-format.
```

Løsning 14

```
F1)   743  =  0000 0010 1110 0111 (2)   =   02E7 (16)

F2)   313 =        0000 0001 0011 1001 (2)
      1-kompl.     1111 1110 1100 0110 (2)
      +                              1 (2)
      ----------------------------------
      -313 =       1111 1110 1100 0111 (2)  =  FEC7 (16)

F3)   200 =        0000 0000 1100 1000 (2)
      1-kompl.     1111 1111 0011 0111 (2)
      +                              1 (2)
      ----------------------------------
      -200 =       1111 1111 0011 1000 (2)  =  FF38 (16)
```

Løsning 15

```
F1)   FF27 (16) = 1111 1111 0010 0111 (2)
        1-kompl.  0000 0000 1101 1000 (2)
        +                            1 (2)
      ----------------------------------
        2-kompl.  0000 0000 1101 1001 (2)  =  00D9 (16)
      FF27 (16) = -00D9 (16) = - ( (13*16) + 9 ) = -217

F2)   5*256 + 6*16 + 4 =1380

F3)   8234 (16) = 1000 0010 0011 0100 (2)
        1-kompl.  0111 1101 1100 1011 (2)
        +                            1 (2)
      ----------------------------------
        2-kompl.  0111 1101 1100 1100 (2)  =  7DCC (16)
      8234 (16) = -7DCC (16)
                = - ((7*4096)+(13*256)+(12*16)+12)
                = -32204
```

Løsning 16

```
A)    A  = 515  =  0000 0010 0000 0011 (2)
      B  = 519  =  0000 0010 0000 0111 (2)
                   -----------------------
      C =          0000 0100 0000 1010 (2)

        =  40A (16) = 1034 (10)

B)    515       =  0000 0010 0000 0011 (2)
      1-kompl.     1111 1101 1111 1100 (2)
      +                              1 (2)
      ----------------------------------
      A = -515  =  1111 1101 1111 1101 (2)
      B =   65  =  0000 0000 0100 0001 (2)
                   -----------------------
      C =          1111 1110 0011 1110 (2)  (negativt)

                   0000 0001 1100 0001 (2)
                +                      1 (2)
                -------------------------
                   0000 0001 1100 0010 (2)
      = -01C2 (16) = -450 (10)
```

Løsning 17

```
A)    B = 519    = 0207 (16) = 0000 0010 0000 0111 (2)
      fortegn vendes
                    1-kompl.   1111 1101 1111 1000 (2)
                    +                             1 (2)
                    ----------------------------------
      -B =          2-kompl.   1111 1101 1111 1001 (2)
      +
      A = 515    = 0203 (16) = 0000 0010 0000 0011 (2)
                               -----------------------
      C =                      1111 1111 1111 1100 (2)
      result. negativ vend
                    1-kompl.   0000 0000 0000 0011 (2)
                    +                             1 (2)
                    ----------------------------------
                    2-kompl.   0000 0000 0000 0100 (2)
        = - 0004 (16) = -4

B)    B = 65     = 0041 (16)   0000 0000 0100 0001 (2)
      fortegn vendes
                    1-kompl.   1111 1111 1011 1110 (2)
                    +                             1 (2)
                    ----------------------------------
      -B =          2-kompl.   1111 1111 1011 1111 (2)
      +
      A = 515    = 0203 (16) = 0000 0010 0000 0011 (2)
                               -----------------------
      C =                   (1)0000 0001 1100 0010 (2)
        = 01C2 (16) = 450

C)    B = -65    = FFBF (16)   1111 1111 1011 1111 (2)
      fortegn vendes
                    1-kompl.   0000 0000 0100 0000 (2)
                    +                             1 (2)
                    ----------------------------------
      -B =          2-kompl.   0000 0000 0100 0001 (2)
      +
      A = -515   = FDFD (16) = 1111 1101 1111 1101 (2)
                               -----------------------
      C =                      1111 1110 0011 1110 (2)
      result. negativ vend
                    1-kompl.   0000 0001 1100 0001 (2)
                    +                             1 (2)
                    ----------------------------------
                    2-kompl.   0000 0001 1100 0010 (2)
        = - 01C2 (16) = -450

D)    B = 32000 = 7D00 (16)   0111 1101 0000 0000 (2)
      fortegn vendes
                    1-kompl.   1000 0010 1111 1111 (2)
                    +                             1 (2)
                    ----------------------------------
      -B =          2-kompl.   1000 0011 0000 0000 (2)
      +
      A =-32000 = 8300 (16) = 1000 0011 0000 0000 (2)
                               -----------------------
      C =                   (1)0000 0110 0000 0000 (2)
        = 0600 (16) = 1536 (overflow)
```

Løsning 18

```
F1)   0000 0000 0000 1010 (2)   =   000A (16)
F2)   0000 0001 0011 1101 (2)   =   013D (16)

F3)   23 =          0000 0000 0001 0111
      1-kompl.      1111 1111 1110 1000
      +                               1
      -------------------------------
      -23 =         1111 1111 1110 1001 (2)   =   FFE9 (16)
```

Løsning 19

a) FF12 (16) = 1111 1111 0001 0010 (2) (negativ)

```
  1-kompl.  0000 0000 1110 1101 (2)
  +                           1 (2)
-----------------------------------
  2-kompl.  0000 0000 1110 1110 (2)  =  00EE (16)
```

FF12 (16) = -00EE (16) = - ((14*16) + 14) = **-238**

b) 2*256 + 3*16 + 5 = **564**

c) 1*4096 + 2*256 + 3*16 + 4 = **4660**

d) 8001 (16) = 1000 0000 0000 0001 (2) (negativ)

```
  1-kompl.  0111 1111 1111 1110 (2)
  +                           1 (2)
-----------------------------------
  2-kompl.  0111 1111 1111 1111 (2)  =  7FFF (16)
```

8001 (16) = -7FFF (16)
= - ((7*4096) + (15*256) + (15*16) + 15)
= **-32767**

Løsning 20

A) B = 47 = 002F (16) = 0000 0000 0010 1111 (2)
fortegn vendes

```
                1-kompl.   1111 1111 1101 0000 (2)
                +                             1 (2)
                ----------------------------------
-B =            2-kompl.   1111 1111 1101 0001 (2)
+
A = 120    = 0078 (16) = 0000 0000 0111 1000 (2)
                         -----------------------
C =                      0000 0000 0100 1001 (2)
```

Løsning 21

```
a)  1 1101 (2)        =  29
b)  11 0101 (2)       =  53
c)  1 0100 0101 (2)   =  325
e)  1100 1110 , 1     =  12,90625
f)  10 , 011 011      =  2,421875
```

Løsning 22

```
a)  1027    =  100 0000 0011 (2)
b)  51202   =  1100 1000 0000 0010 (2)
c)  678     =  10 1010 0110 (2)
d)  1412    =  101 1000 0100 (2)
e)  0,78    =  0 , 11001 (2)  med 5 dec. oprundet
f)  14,13   =  1110 , 0010001 (2)  med 7 dec. oprundet
```

Løsning 23

```
a)  113 (16)   =  275
b)  BEA (16)   =  3050
c)  E04 (16)   =  3588
d)  FABE (16)  =  64190
```

Løsning 24

a)	14	=	0000 1110 (2)	
b)	-14	=	1111 0010 (2)	
c)	101	=	0110 0101 (2)	
d)	-101	=	1001 1011 (2)	
e)	-56	=	1100 1000 (2)	
f)	250	=	1111 1010 (2)	tal uden fortegn

Løsning 25

a)	00 00 1 110 (2)	=	0E (16)	=	016 (8)
b)	11 11 0 010 (2)	=	F2 (16)	=	362 (8)
c)	01 10 0 101 (2)	=	65 (16)	=	145 (8)
d)	10 01 1 011 (2)	=	9B (16)	=	233 (8)
e)	11 00 1 000 (2)	=	C8 (16)	=	310 (8)
f)	11 11 1 010 (2)	=	FA (16)	=	372 (8)

Løsning 26

031 (8) = 25 424 (8) = -236
313 (8) = 203 717 (8) = -49

031 (8) = 000 011 001 (2) = 0 0001 1001 (2) = 019 (16)

424 (8) = 100 010 100 (2) = 1 0001 0100 (2) = 114 (16)

313 (8) = 011 001 011 (2) = 0 1100 1011 (2) = 0CB (16)

717 (8) = 111 001 111 (2) = 1 1100 1111 (2) = 1CF (16)

Løsning 27

a) 432 = integer = 01 B0 (16), ascii= 34 33 32 (16)
b) -432 = integer = FE 50 (16), ascii= 2D 34 33 32 (16)

Løsning 28

	Decimal	Binær	Octal	Hex
a)	1777	0000011011110001	003361	06F1
b)	549	0000001000100101	001045	0225
c)	-615	1111110110011001	176631	FD99
d)	2705	0000101010010001	005221	0A91
e)	1349	0000010101000101	002505	0545
f)	-1776	1111100100010000	174420	F910
g)	2994	0000101110110010	005662	0BB2
h)	4025	0000111110111001	007671	0FB9
i)	4407	0001000100110111	010467	1137
j)	1153	0000010010000001	002201	0481

2.

Operativsystemets brugerfaciliteter

Opgaver

Indhold:

2.1.a

Basale DOS kommandoer
Opgaver

Bjørk Busch

Opgave BDOS001.

Ved brug af **DIR** kommandoen skal du undersøge følgende:

1) Hvor mange filer findes der på drev C i root-directory (C:\).
2) Hvilken dato er filen AUTOEXEC.BAT sidst rettet og hvor meget fylder den.
3) Findes der en fil på drev C eller D en fil med navnet NIBBELS.BAS, og hvis der gør i hvilket katalog evt. underkatalog.
4) Find de filer der ligger i kataloget DOS, som starter med A og hedder COM som extension.
5) Hvor mange filer i DOS har B som andet bogstav.

Opgave BDOS002.

Ved brug af **MD** skal du på DIN DISKETTE oprette følgende kataloger(hvis disketten ikke er formateret skal dette først gøres).

1) I rooten oprettes WP, MP og VI.
2) I kataloget WP oprettes to underkataloger BREVE og SANGE.
3) I underkataloget BREVE oprettes tre under-kataloger MOR, FAR og BBB.
4) I kataloget MP oprettes to underkataloger INGE og BBB.

Kontrollér med kommandoen **TREE**

Opgave BDOS003.

Opret følgende filer med en lille tekst, ved brug af editoren **EDIT**, TurboPascal's editor (**TP**) eller med:

COPY CON <*filnavn.ext*>

Der afsluttes med **CTRL Z**

Hvor det ønskede filnavn indsættes som <*filnavn.ext*>.

Du kan have glæde af kommandoen CD :

1) A:\WP\PENGE.ASC
2) A:\WP\HJUL.ASC
3) A:\WP\BREVE\BREV1.TXT
5) A:\WP\BREVE\MOR\BREV1.TXT
6) A:\WP\BREVE\MOR\HJEMVE.ASC
7) A:\MP\INGE\OPG31.ASC
8) A:\MP\BBB\OPG480.TXT
9) A:\INGEN.SU

kontrollér indhold af disketten med **DIR** og **TREE**.

Du kan se af ASCII-tekstfiler med kommandoen **TYPE**, som evt. kan kombineres med **MORE** (se bog/manual).

Opgave BDOS004.

Du skal nu kopiere følgende filer:

1) PENGE.ASC skal kopieres til underkataloget FAR.
2) INGEN.SU kopieres til WP.
3) Alle filer fra WP kopieres til MOR.
4) BREV1.TXT kopieres til MOR (hvad sker der med det "gamle" brev i MOR)
5) "Stil" dig i MOR og kopier filen BREV1.TXT så du får to mere med navnene BREV1A.TXT og BREV2.TXT.
6) Stil dig i kataloget BBB under MP og kopier filen fra INGE idet den skal hede ISOPG31.TXT.
7) Kopier herefter alle filer fra BBB under MP til BBB under BREVE.

Kontrollér indhold med **CD**, **DIR** og evt. **TREE**.

Opgave BDOS005.

Du skal nu foretage ændringer af filnavne:

1) Omdøb filen ISOPG31.TXT til INGE31.TXT.
2) Omdøb alle filer i WP med extension WP til extension WP5.
3) Omdøb alle filer i MOR som starter med BREV til LETTER hvor resten af navnet skal være uændret.

Opgave BDOS006.

Du skal nu slette nogle filer og kataloger:

1) Slet filen INGE31.TXT.
2) Slet alle filer i MOR som har extension TXT.
3) Fjern kataloget FAR.
4) Fjern kataloget MOR.
5) Slet alle filer som ligger i WP (hvad sker der med underkataloger).
6) Foretag det fornødne for at fjerne WP kataloget.

Opgave BDOS007.

Du skal nu kopiere disketten ved brug af DISKCOPY. Slå den op i bogen.(BEMÆRK: Det er meningen at du skal kopiere fra A: til A: !!)
Husk at man skal være varsom, da modtageren bliver identisk med afsender, således at en tom diskette der kopieres til en med indhold, vil give to tomme.

Opgave BDOS008.

Du skal nu prøve XCOPY:

1) Kopier kataloget MP incl. filer og underkataloger over i kataloget IV.
2) Kopier hele diskettens indhold over på en anden diskette (husk hvis det er en ny diskette skal den først formateres).
2) Hvordan kan man undgå at overskrive eksisterende filer ?
3) Er der andre smarte muligheder med XCOPY.

2.1.b

Udvidede DOS kommandoer
Opgaver

Bjørk Busch

Opgave UDOS001.

Hvilken forskel er der på kommandoerne

DIR /P og **DIR | MORE** ?

Det er en ganske speciel forskel der ønskes som svar.

Opgave UDOS002.

Opret et **bibliotek på disketten** og kopier nogle filer derover.
Flyt dig over i biblioteket, så det bliver det aktuelle bibliotek.

Hold tungen lige i munden og afprøv følgende

DEL *.*

Hvad er effekten ?

Kopier filer over i biblioteket igen og prøv nu

ECHO Y | DEL *.*

Hvad er effekten ? Er der forskel fra før ?

Opgave UDOS003.

I denne opgave skal du ændre DOS-PROMPT'en, der normalt viser aktuelt drev og katalognavn efterfulgt af et større end tegn.

a) Prøv kommandoen **PROMPT $d** hvad sker der ?
b) Sæt prompten, så der skrives **KL. xx.xx.xx** efterfulgt af den normale prompt (xx.xx.xx er det aktuelle klokkeslet).
c) Hvad gør **PROMPT $e[4m$p$g** (skal være lille e)
d) Hvad gør **PROMPT $e[5m$p$g** (skal være lille e)
e) Hvad gør **PROMPT $e[0m$p$g** (skal være lille e)
f) Hvad gør **PROMPT $e[4;5m$pge[0;7m** (skal være lille e)

I a-f anvendes de såkalde ANSI-escape sekvenser (se evt. andre muligheder i dosbog/manual).

Opgave UDOS009.

Fremstil en oversigt over filer i aktuelt katalog, der er sorteret på følgende:

a) Filnavn
b) extention
c) dato
d) størrelse

Prøv både med brug af DIR med parametre og med DIR uden parametre sammen med SORT.
Kan alle oversigter laves på begge måder ?

Opgave UDOS010.

Fremstil ved brug af DIR og FIND en oversigt over filer fra en bestemt dato.

Opgave UDOS011.

Beskriv hvorledes harddisken på DEC-maskinerne er opbygget med hensyn til logiske drev og kataloger.
Der ønskes herunder en redegørelse for størrelsen af harddisken, og for de enkelte drevs størrelse, samt hvor meget plads, der er tilbage på hvert drev.
Hvad hedder de(t) logiske drev, der er knyttet til netværket og beskriv katalogstrukturen.
NB! Pas på med kommandoer, der kan ødelægge indholdet af disken, så du ikke svarer ja til opdatering eller sletning.
Speceielt relevante kommnandoer er CHKDSK, FDISK, TREE og DIR.

Opgave UDOS012.

Opret en tekstfil på din diskette og gør den skrivebeskyttet med ATTRIB.
Prøv at slette filen samt give den et nyt navn.

Opgave UDOS013.

Du skal formattere en system-diskette ved brug af FORMAT kommandoen.
Undersøg først indhold med DIR.
Kontrollér indhold af disketten med kommandoen CHKDSK. Hvad får du at vide ?
Prøv herefter CHKDSK med /V som parameter samt drev og *.*.
Du kan nu se navnene på de skjulte filer.
Hvad hedder systemfilerne på den anvendte maskine ?

Opgave UDOS014.

Du skal nu prøve, at starte maskinen op på disketten.

Varmstart kan foretages ved at trykke CTRL, ALT og DEL.

Virker tastaturet som det plejer (er f.eks. de rigtige tegn på tasterne *æøå*) ?

Kan du benytte kommandoerne DIR, COPY og DEL ?

Hvad med TREE og CHKDSK ?

Hvad kan forklaringen være ?

Hvorfor virker nogle kommandoer og ikke andre ?

Opgave UDOS015.

Du skal nu prøve, at fremstille samme systemdiskette som før, men denne gang skal disketten forfatteres som datadiskette, og systemet overføres med SYS og evt. COPY, hvis der mangler filer (COMMAND.COM).

Husk at maskinen først skal startes fra harddisk igen for at virke "normalt".

2.2.a

DOS batch programmering
Opgaver

Bjørk Busch

Udgangspunktet for alle opgaverne er, at rutinerne også skal kunne afvikles under MSDOS 5.0.
Flere af rutiner kan klares simplere fra MSDOS 6.2, undersøg mulighederne.

Opgave BAT001.

Fremstil et BAT-program, der ved brug af DIR og FIND en oversigt over BAK filer i det aktuelle katalog.

Opgave BAT002.

I BATCH-programmer er det sommetider nødvendigt at kunne give en meddelelse til brugeren. Det kan gøres med ECHO-kommandoen.
Lav et BATCH-program der kan udskrive en ledetekst på skærmen.

Opgave BAT003.

I fortsættelse af foregående opgave kan du prøve at sætte en linie ind øverst i dit BATCH-program der ser ud som følger,

```
@ECHO OFF
```

Hvilken effekt har denne linie ?

Opgave BAT004.

Prøv at indsætte kommandoen PAUSE i dit BATCH-program.
Hvad er effekten ?

Opgave BAT005.

Meddelelsen fra PAUSE kan erstattes på en mere elegant måde. Teksten som PAUSE udskriver kan redirigeres til "et tomt hul" med følgende kommando

```
PAUSE > NUL
```

Hvad er effekten ? Hvortil kunne denne redirigering være nyttig ?

Opgave BAT006.

Fremstil ved brug af DIR, FIND og MORE et BAT-program, der giver en oversigt over filer på C drevet, der IKKE har extension BAT, EXE og COM. Det kan blive nødvendig med flere FIND.

Opgave BAT007.

Fremstil et BAT-program "**VISPAS**.BAT", der ved brug af DIR og FIND en oversigt over filer med extention PAS fra en bestemt dato, idet datoen skal kunne angives som parameter.

Eks. p start **VISPAS 05-08-93**

Opgave BAT008.

Udvid BAT-programet fra før så der også skal angives katalognavn som parameter.

Eks. på start **VISPAS 01-08-93 A:\PASOPG**

Opgave BAT009.

Udvid BAT-program fra før så der udskrives en vejledning hvis der ikke angives 2 parametre.

Opgave BAT010.

Lav et BAT-program der kan omdøbe et katalog. Programmet skal hedde NEW-NAME.BAT og kaldes på følgende måde

NEWNAME <oldcat> <newcat>

f.eks.

NEWNAME WP GP

Opgave BAT011.

Vi har afprøvet | (pipe), men der er mange andre muligheder. V.h.a. kommandoen CHKDSK (check disk) og FIND-kommandoen kan man lave en fil-søger.
Lav et BAT-program (LOOKFOR.BAT) der kan søge efter filen der er angivet som parameter 1.

Et kald kunne se således ud

LOOKFOR <filnavn.ext)

f.eks.

LOOKFOR TURBO.EXE

Opgave BAT012.

Fremstil et BAT-program, der giver oversigten fra foregående opgave, idet katalognavnet og datoen angives som parameter og oversigten dels skal gemmes i en fil, dels skal vises på skærmen (en side af gangen)

Opgave BAT013.

Fremstil ved brug af ATTRIB og FIND, en oversigt over skjulte filer.

Opgave BAT014.

Fremstil ved brug af CHKDSK/DIR, FIND og SORT et BAT-program der giver en sorteret oversigt over alle filer med extension EXE, COM og BAT på såvel C: som D:
Det bliver nødvendigt at opbygge en fil trinvis med det udvalgte og sortere den til sidst. Det skal bemærkes at > > kan anvendes når man ved redirektion ønsker at bygge videre på en fil uden at den slettes (append). Hvis filen ikke findes bliver den oprettet.

Opgave BAT015.

Skriv et BAT-program (KOPI) der kan kopiere en diskette uden underkataloger ved følgende metode:

- Opret kataloget FLOPCOPY på disk
- Kopier alle filer fra disketten til kataloget
- Vent på at brugeren skifter diskette og trykker på en tast.
- Kopier filer tilbage til disketten.
- Slet FLOPCOPY

Diskens navn angives som parameter.

Opgave BAT016.

Fremstil et BAT-program, der ved brug af FOR og DIR, kan give en oversigt over alle filer i aktuelt katalog med extension EXE, COM og BAT.

Opgave BAT017.

Programmet fra foregående opgave udbygges, således at oversigten dannes i en ny fil, der så vises sorteret på skærmen.
Programmet skal rydde op efter sig.

Opgave BAT018.

Programmet fra foregående opgave skal nu udbygges, således at der kan angives et katalog som parameter.

Opgave BAT019.

Programmet udbygges yderligere, således at der kan angives flere kataloger som parametre.
Hvis der angives parameteren /?, /h eller /H skal der udskrives en lille vejledning i stedet for en filoversigt.

Opgave BAT020.

Fremstil nu et simpelt MENU-system med følgende punkter:

1. Udskrift af menu-billede og aktivering af menu-prompt.
2. Opstart af WP51.
3. Opstart af PARADOX.
4. Forlad menu.

Menu-systemet fremstilles ved at oprette en BAT-fil for hver af punkterne (1.BAT , 2.BAT ...)
Du kan evt. se hvorledes de anviste produkter startes op på den skolens installation.

Menu-prompten opsættes ved brug af ANSI escape-sekvenser og skal fungere på følgende måde:
Der udskrives i nederste linie med anden farve "Tryk 1 for MENU" og CURSOR placeres herefter øverste venstre hjørne, hvor der fortsættes med "normal prompt".

Ved forlad menu skal skærmen blankes og prompt ændres til normal.

Opgave BAT021.

Du bedes fremstille en komandofil FLYT.BAT, som kan flytte en eller flere filer fra et katalog til et andet. **Du må ikke anvende MOVE, da bat-programmet også skal kunne køre på en maskine med MSDOS 5.0.**

Kommandofilen FLYT.BAT skal opstartes på følgende måde:

FLYT *frakat tilkat fil [fil .....]*

1. parameteren "frakat" angiver:
   det katalog filen skal flyttes fra.

2. parameteren "tilkat" angiver:
   det katalog filen / filerne skal flyttes til

3. parameteren og de efterfølgende "fil" angiver:
   filnavne på de filer der skal flyttes (excl. katalog). Der tale om enkeltfiler, således at der ikke angives wildcard (* og ?).

Eks. på hvorledes rutinen skal kunne anvendes:

FLYT A:\KAT1 A:\KAT2 MINFIL

Resultat:
Filen A:\KAT1\MINFIL flyttes til A:\KAT2\MINFIL
Filen A:\KAT1\MINFIL slettes

FLYT C:\KAT1 A:\KAT2 FIL1 FIL2 FIL3

Resultat:
Filen C:\KAT1\FIL1 flyttes til A:\KAT2\FIL1
Filen C:\KAT1\FIL1 slettes
Filen C:\KAT1\FIL2 flyttes til A:\KAT2\FIL2
Filen C:\KAT1\FIL2 slettes
Filen C:\KAT1\FIL3 flyttes til A:\KAT2\FIL3
Filen C:\KAT1\FIL3 slettes

BAT-programmet FLYT.BAT skal opbygges på følgende måde:

De kontroleres først, at der er mindst 3 parametre (3. parameteren ikke er tom).
Hvis der ikke er mindst 3 parametre skal der udskrives en kort (meget kort) vejledning i brugen rutinen og der skal herefter stoppes.

Hvis der var mindst 3 parametre fortsættes med flytningen af filerne, idet dette skal ske i en løkke, hvor der flyttes en fil af gangen.

I forbindelse med løkken vil være nødvendigt at rokkere parametrene og løkken standses, når der ikke er flere filer (den aktuelle fil-parameter er TOM).
Ved rokkeringen af parametrene bliver fra- og tilkatalog ikke tilgængelige som parametre. Det er derfor nødvendigt at gemme dem inden opstart af løkken (der henvises til note og DOS bog om kommandoer i bat-programmer).

I løkken foretages flytningen på følgende måde:

1) Først testes det om det nye filnavn allerede findes, hvis dette er tilfældet udskrives en fejlmeddelelse og filen flyttes ikke, men der fortsættes med den næste fil (hvis flere).

2) Hvis det nye filnavn (tilkat\fil) ikke findes, kontrolers at fra-filen findes (frakat\fil). Hvis dette ikke er tilfældet udskrives fejlmeddelelse og der fortsættes med den næste fil (hvis flere).

3) Hvis alt var OK kopieres filen til det nye navn.

4) Hvis kopieringen gik godt (den nye fil er oprettet) slettes den gamle, ellers udskrives en fejlmeddelelse. I begge tilfælde fortsættes herefter med evt. næste fil.

Opgave BAT022.

Der skal fremstilles et BAT-program MULTIPRN, som kan udskrive en eller flere filer fra et eller flere kataloger.

Kommandoen MULTIPRN skal opstartes på følgende måde:

MULTIPRN *extention [path] [path] [.....]*

1. parameteren "extention" angiver:
Extention på de ascii-tekst-filer der skal udskrives.
Det **kan forlanges**, at det er skrevet med **store bogstaver** (behøver ikke virke hvis ikke).
Kun følgende extention er gyldige:
TXT, PAS og ASM

2 - ? parameterene "path" angiver:
Kataloger (evt. incl. drev) hvor filerne, der skal udskrives findes i.
Der kan angives flere kataloger.
Hvis der ikke angives et katalog anvendes aktuelt.
Hvis der angives et katalog SKAL AKTUELT katalog angives (evt. som .) hvis det skal medtages.

Eks. på hvorledes rutinen skal kunne anvendes:

MULTIPRN TXT

Resultat: Alle filer med extention TXT i aktuelt katalog udskrives.

MULTIPRN TXT C:\PASCAL A:\OPG

Resultat: Alle filer med extention TXT i katalogerne C:\PASCAL og A:\OPG udskrives.

Det **skal** kontrolleres, at der er en parameter angivet. Hvis der ikke er, skal der udskrives en vejledning og udførslen afsluttes.

Det **skal** kontrolleres, at 1. parameteren (extention) er et af de godkendte (TXT, PAS eller ASM). Hvis dette ikke er tilfældet, skal der udskrives en passende fejlmeddelelse og udførslen afsluttes.

Hvis der ikke er angivet et katalog (kun een parameter) skal **aktuelt** katalog behandles, ellers behandles **de angivne** kataloger og kun disse et efter et.

Behandling pr. katalog:

Hvis der ingen filer findes med det angivne extention i kataloget, så udskrives der en meddelelse på skærmen og der fortsættes med behandling af evt. næste angivne katalog.

Hvis der findes filer med det angivne extention i kataloget, så behandles filerne en efter en.

Behandling pr. fil:

I DENNE DEL SKAL UDSKRIFTER **IKKE FORETAGES DIREKTE PÅ PRINTEREN**, DA DET BRUGER FOR MEGET PAPIR. DU KAN ISTEDET SENDE ALLE UDSKRIFTER TIL SKÆRMEN (brug **TYPE og MORE** istedet for PRINT og brug **CON:** istedet for PRN:).
Angiv i REM sætning hvad der skal ændres i produktions udgave.

Først udskrives en meddelelse på skærmen, om at filen sendes til udskrift. Denne skal indeholde filnavnet.

Herefter udskrives en linie med filens navn på "printeren".

Herefter udskrives en linie med bindestreger på "printeren".

Herefter udskrives selve filen på "printeren".

Endelig udskrives en meddelelse på skærmen, om at filen er afsendt. Denne skal ligeledes indeholde filnavnet.

TIPS:

Extention kan fastholdes i en systemvariabel (se note og DOS bog).
Lav et særskilt BAT-program, der kan behandle een fil, der angives som parameter. Dette BAT-program kan så kaldes fra "hoved" BAT-programmet.
Der kan evt. opbygges en printfil først, som så sendes ud samlet på "printeren" (i testudgave skærmen).

I hovedprogrammet kan man se på FOR-konstruktionen.

2.3.a

DOS systemtilpasning
Opgaver

Opgave SYS001.

Formater en SYSTEMDISKETTE.

Du skal nu prøve at gøre maskinen lidt mere normal, når der startes på diskette. I første omgang skal vi lave en fil, som hedder CONFIG.SYS, hvor vi kan give DOS informationer, for tilpasning af systemet.
Denne fil skal være en alm. tekst-fil(ASCII-FIL), som kan fremstilles med en valgfri editor (eller COPY CON).

CONFIG.SYS skal tilpasses med følgende (som du kan slå op i DOS-håndbogen eller du kan kigge CONFIG.SYS på C:\):

COUNTRY
ANSI.SYS
DISPLAY.SYS
SHELL

Prøv at starte maskinen fra disketten. Er der forskelle fra den "tomme" system-diskette ?
HUSK DER MÅ IKKE ÆNDRES PÅ HARDDISKEN.

Opgave SYS002.

Vi skal nu udbygge opstarten yderligere med en tekst-fil, som hedder AUTOEXEC.BAT, hvor vi kan angive kommandoer, som skal udføres i forbindelse med opstarten.
Kommandoerne skulle ellers indtastes fra tastaturet ved hver opstart.

AUTOEXEC.BAT skal kunne klare følgende:

Dansk tegnsæt på skærm.
Dansk tegnsæt på tastatur.
Prompt skal indeholde drev, katalog og >.
Søgesti for programmer skal svare til den, som anvendes ved opstart på harddisk.

HUSK DER MÅ IKKE ÆNDRES PÅ HARDDISKEN.

Ved løsningen af de sidste to opgaver, må du gerne se hvorledes filerne CONFIG.SYS og AUTOEXEC.BAT ser ud på harddisken, men du skal ikke lave en identisk opstart med menu og lignende, men kun det i opgaven skitserede.
Det er ikke meningen at netværket skal etableres.

Opgave SYS003.

Beskriv de enkelte linier i filerne CONFIG.SYS og AUTOEXEC.BAT på skolens DEC-maskiner.

Opgave SYS004.

Prøv at oprette en RAM-disk på ca. 200 kb med RAMDRIVE.SYS (opsættes i CONFIG.SYS - husk herefter at reboote).
Kopier et mindre program over på RAM-disken og start herefter programmet op fra dette nye "drev".
Hurtigt ikke !!!

Opgave SYS005.

Undersøg ved brug af MEM kommandoen, hvor meget hukommelse der er tilbage når hele operativsystemet er indlæst.
Hvor meget er der tilbage når du starter op fra diskette ?

Opgave SYS006.

Fremstil en opstartsdiskette.

Der skal foretages de normale opsætninger af CONFIG.SYS og AUTOEXEC.BAT.

Ved afslutningen af AUTOEXEC.BAT, skal der udskrives et velkomstbillede på skærmen, idet der (evt. kan anvendes forskellige farver ved brug af ANSI-escape-sekvenser).

ESC værdien kan indtastes i editoren ved brug af "ALT-tricket":

> Hold på ALT-tasten og indtast ascii-værdien på det numeriske tastatur (ESC = 27) og slip herefter ALT-tasten.

Opgave SYS007.

Opstartsdisketten tilpasses nu med følgende:

- Der ønskes så meget som muligt af den konventionelle hukommelse (de første 640 kb ram) som muligt.
- Der ønskes ikke anvendt EMS hukommelse men kun XMS.
- Der ønskes diskcache (smartdrv) samt en virtuel disk på 500 kb (ramdrive).
- Disketten skal ved opstarten kopieres over på ramdisken og systemet skal omdirigeres således, at ramdisken kommer til at hedde b:

3.

Filsystemer

Opgaver

Indhold:

3.1.a

Bootsector, fattabel og directory opbygning
Opgaver

Bjørk Busch

Opgave FIL001.

På disketter og harddiske findes en såkaldt BOOT-SECTOR, der beskriver mediet.

Nedenstående er en udskrift af en sådanne.

Udskriften er i HEX-dump format og startadressen på hver linie står yderst til venstre.

HEX-DUMP AF BOOT-SECTOR:

```
ADR :   0  1  2  3  4  5  6  7    8  9  A  B  C  D  E  F
0000:  -- -- -- 49 42 4D 20 20 - 34 2E 30 00 02 01 01 00
0010:  02 E0 00 60 09 F9 07 00 - 0F 00 02 00 00 00 -- --
```

1) I Adresse 0003-000A står producentnavnet i ASCII format. Hvad hedder producenten ?

2) I Adresse 000B-000C står hvor mange bytes, der er pr. sector. Informationen er et 16-bits binært tal uden fortegn og den mindst betydende byte (8-bit) står først. Hvor mange bytes er der pr. sector ?

3) I Adresse 0011-0012 står hvor mange filer der er plads til i ROOT-DIRECTORY. Informationen er et 16-bits binært tal uden fortegn og den mindst betydende byte (8-bit) står først.
Hvor mange filer er der plads til ?

4) I Adresse 0013-0014 står hvor mange sectorer der er ialt på mediet. Informationen er et 16-bits binært tal uden fortegn og den mindst betydende byte (8-bit) står først. Hvor mange sectorer er der ialt på mediet?
Hvad er den samlede kapacitet så på mediet.

5) I Adresse 0018-0019 står hvor mange sectorer der er på hver side. Informationen er et 16-bits binært tal uden fortegn og den mindst betydende byte (8-bit) står først. Hvor mange sectorer er der pr side?

Opgave FIL002.

Nedenstående vise en udskrift af informationer om en enkelt fil hentet fra ROOT-KATALOGET.

Udskriften er i HEX-dump format og startadressen på hver linie står yderst til venstre.

HEX-DUMP AF kataloginformationer for en enkelt fil:

```
ADR :   0  1  2  3  4  5  6  7    8  9  A  B  C  D  E  F

0000:  54 55 52 42 4F 20 20 20 - 45 58 45 20 00 00 00 00
0010:  00 00 00 00 00 00 00 20 - 71 0F 02 00 48 C2 01 00
```

1) I Adresse 0000-0007 står filnavnet i ASCII format.
Hvad hedder filen ?

2) I Adresse 0008-000A står filextention i ASCII format.
Hvad er filens extention ?

3) I Adresse 0016-0017 står files tidspunkt.
Tidspunktet er på 16 BIT, hvor den mindst betydende BYTE (8 BIT) står først.
Tidspunktet opbevares på følgende måde:
Bit 0 - 4 er sekunder/2 (skal ganges med 2)
Bit 5 - 10 er minutter
Bit 11 - 15 er timer
Hvad er filens tidspunkt ?

4) I Adresse 0018-0019 står filens dato.
Datoen er på 16 BIT, hvor den mindst betydende BYTE (8 BIT) står først.
Datoen opbevares på følgende måde:
Bit 0 - 4 er dag
Bit 5 - 8 er måned
Bit 9 - 15 er årstal-1980 (adder 1980 til)
Hvad er filens dato ?

5) I Adresse 001C-001F står filens størrelse som et 32 bits tal, hvor de mindst betydende bytes står først (baglæns).
Hvar stor er filen ?

Opgave FIL003.

Nedenstående vise en udskrift af informationer om 3 filer hentet fra ROOT-KATALOGET.

Udskriften er i HEX-dump format og startadressen på hver linie står yderst til venstre.

HEX-DUMP AF kataloginformationer:

```
ADR :   0  1  2  3  4  5  6  7    8  9  A  B  C  D  E  F
```

1. fil's informationer

```
0000:  41 20 20 20 20 20 20 20 - 42 41 54 20 00 00 00 00
0010:  00 00 00 00 00 00 53 0F - 3A 0F 02 00 1C 00 00 00
```

2. fil's informationer

```
0020:  41 53 54 52 4F 4C 38 20 - 44 4F 43 20 00 00 00 00
0030:  00 00 00 00 00 00 23 A5 - F2 0C 03 00 80 08 00 00
```

3. fil's informationer

```
0040:  41 53 54 52 4F 4C 38 31 - 45 58 45 20 00 00 00 00
0050:  00 00 00 00 00 00 B2 58 - F4 0C 06 00 B8 FB 01 00
```

Informationerne om hver fil er opbygget som i opgave 2:

Filnavn	8 bytes	i ascii
Filextention	3 bytes	i ascii
Attribute	1 byte	(1 bit pr. kendetegn - omtales senere)
Reserveret	10 bytes	
Tidspunkt	2 bytes	(time, minut, sek.) se opgave 2
Dato	2 bytes	(år, måned, dag) se opgave 2
Filstart	2 bytes	(16 bit tal - omtales senere)
Filstørrelse	4 bytes	(32 bit tal se opgave 2)

1) Hvad hedder de 3 filer (filnavn og extention) ?

2) Hvad er filernes opdateringstidspunkt ?

3) Hvad er filernes opdateringsdato ?

4) Hvar store er filerne ?

Opgave FIL004.

Givet er følgende DUMP fra en diskette:

BOOT-SECTOR:

Drev:A Dos Sector: 0 Bios Spor: 0 Side: 0 Sector: 1

ADR :	0	1	2	3	4	5	6	7		8	9	A	B	C	D	E	F
0000:	EB	40	90	46	4C	20	76	31	-	2E	32	31	00	02	04	01	00
0010:	02	90	00	A0	05	F9	02	00	-	09	00	02	00	00	00	00	00

FAT-TABEL:

Drev:A Dos Sector: 1 Bios Spor: 0 Side: 0 Sector: 2

ADR :	0	1	2	3	4	5	6	7		8	9	A	B	C	D	E	F
0000:	F9	FF	FF	03	40	00	05	60	-	00	FF	8F	00	09	A0	00	FF
0010:	0F	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00

ROOT-DIRECTORY:

Drev:A Dos Sector: 5 Bios Spor: 0 Side: 0 Sector: 6

ADR :	0	1	2	3	4	5	6	7		8	9	A	B	C	D	E	F
0000:	54	48	45	4C	50	20	20	20	-	43	4F	4D	21	00	00	00	00
0010:	00	00	00	00	00	00	61	80	-	12	17	02	00	B8	26	00	00
0020:	4C	41	42	50	43	44	20	20	-	20	20	20	28	00	00	00	00
0030:	00	00	00	00	00	00	40	BB	-	AB	18	00	00	00	00	00	00
0040:	54	48	45	4C	50	20	20	20	-	44	4F	43	02	00	00	00	00
0050:	00	00	00	00	00	00	20	10	-	5D	15	07	00	C3	1D	00	00
0060:	E5	48	45	4C	50	20	20	20	-	43	4F	4D	20	00	00	00	00
0070:	00	00	00	00	00	00	61	80	-	12	17	1F	00	B8	26	00	00
0080:	E5	48	45	4C	50	20	20	20	-	44	4F	43	20	00	00	00	00
0090:	00	00	00	00	00	00	20	10	-	5D	15	24	00	C3	1D	00	00
00A0:	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00
00B0:	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00

a) Fortolk informationerne i BOOT-sektoren.

b) Fortolk informationerne i ROOT-directory.
Hvad er navnet på disketten ?
Hvilke filer er der på disketten.
For filerne ønskes navn, størrelse, dato og tid for sidste ændring, filatribute (readonly, hiden m.m) samt clusternr hvor filen starter.

c) Find ved hjælp af FAT-tabellen ud af, hvor filerne er pladseret på disketten (clustre).
Hvor meget plads optager de enkelte filer fysisk på disketten.
Du kan evt. ved hjælp af informationerne fra BOOT-sectoren prøve at finde den fysiske pladsering af filerne i form af spor, side og sector.

Opgave FIL005.

Efter at den ene fil fra foregående opgave er blevet ændret fås følgende dump.

FAT-TABEL:

```
Drev:A  Dos Sector: 1     Bios Spor: 0 Side: 0 Sector: 2
ADR :    0  1  2  3  4  5  6  7    8  9  A  B  C  D  E  F
0000:   F9 FF FF 03 40 00 05 60 - 00 0B 80 00 09 A0 00 FF
0010:   CF 00 0D F0 FF 00 00 00 - 00 00 00 00 00 00 00 00
```

ROOT-DIRECTORY - data for den ændrede fil:

```
Drev:A  Dos Sector: 5     Bios Spor: 0 Side: 0 Sector: 6
ADR :    0  1  2  3  4  5  6  7    8  9  A  B  C  D  E  F
    :   54 48 45 4C 50 20 20 20 - 43 4F 4D 20 00 00 00 00
    :   00 00 00 00 00 00 C0 BE - A4 18 02 00 BB 3C 00 00
```

a) Fortolk informationerne om den ændrede fil:
Navn, størrelse, dato og tid for sidste ændring, filatribute (readonly, hiden m.m) samt clusternr hvor filen starter.

b) Find ved hjælp af FAT-tabellen ud af, hvor filen er pladseret på disketten (clustre).
Hvor meget plads optager filen på disketten.
Hvor meget kan den udvides uden at forbruge mere plads fysisk på disketten.

Opgave FIL006.

Givet er følgende DUMP fra en diskette/harddisk ?:

BOOT-SECTOR:

ADR :	0	1	2	3	4	5	6	7		8	9	A	B	C	D	E	F
0000:	EB	34	90	4D	53	44	4F	53	-	33	2E	32	00	02	08	01	00
0010:	02	00	02	13	20	F8	04	00	-	11	00	04	00	11	00	00	00

FAT-TABEL:

ADR :	0	1	2	3	4	5	6	7		8	9	A	B	C	D	E	F
0000:	F8	FF	FF	03	40	00	05	60	-	00	FF	8F	00	09	A0	00	0B
0010:	C0	00	0D	F0	FF	0F	00	01	-	11	20	01	13	F0	FF	15	60
0020:	01	FF	FF	FF	FF	FF	FF	FF	-	FF	FF	1D	F0	01	FF	0F	02
0030:	22	F0	FF	23	40	02	25	60	-	02	27	80	02	FF	AF	02	2B
0040:	C0	02	2D	F0	FF	2F	00	03	-	31	20	03	33	40	03	35	60
0050:	03	37	F0	FF	47	F0	FF	3B	-	F0	FF	FF	EF	03	FF	0F	04
0060:	FF	2F	04	43	40	04	FF	FF	-	FF	FF	8F	04	49	A0	04	4B
0070:	C0	04	4D	E0	04	4F	00	05	-	51	20	05	53	40	05	55	60

ROOT-DIRECTORY:

ADR :	0	1	2	3	4	5	6	7		8	9	A	B	C	D	E	F
0000:	49	42	4D	42	49	4F	20	20	-	43	4F	4D	27	00	00	00	00
0010:	00	00	00	00	00	00	60	09	-	96	10	02	00	20	42	00	00
0020:	49	42	4D	44	4F	53	20	20	-	43	4F	4D	27	00	00	00	00
0030:	00	00	00	00	00	00	00	60	-	E7	0C	07	00	40	6F	00	00
0040:	43	4F	4D	4D	41	4E	44	20	-	43	4F	4D	23	00	00	00	00
0050:	00	00	00	00	00	00	00	60	-	75	0C	0E	00	3C	5C	00	00
0060:	48	41	52	44	52	49	56	45	-	53	59	53	23	00	00	00	00
0070:	00	00	00	00	00	00	00	60	-	C6	0E	14	00	1B	27	00	00

a) Fortolk informationerne i BOOT-sektoren.

b) Fortolk informationerne for de første 2 filer i ROOT-directory.
For filerne ønskes navn, størrelse, dato og tid for sidste ændring, filatribute (readonly, hiden m.m) samt clusternr hvor filen starter.

c) Find ved hjælp af FAT-tabellen ud af, hvor første fil er pladseret (clustre).
Hvor meget plads optager filen fysisk.

Opgave FIL007.

Givet er følgende DUMP fra en diskette:

BOOT-SECTOR:

Drev:A Dos Sector: 0 Bios Spor: 0 Side: 0 Sector: 1

ADR :	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	-	*8*	*9*	*A*	*B*	*C*	*D*	*E*	*F*
0000:	EB	3C	90	49	42	4D	20	20	-	34	2E	30	00	02	01	01	00
0010:	02	E0	00	60	09	F9	07	00	-	0F	00	02	00	00	00	00	00
0020:	00	00	00	00	00	00	29	FB	-	10	66	34	48	49	47	48	35
0030:	20	20	20	20	20	20	46	41	-	54	31	32	20	20	20	FA	33

FAT-TABEL:

Drev:A Dos Sector: 1 Bios Spor: 0 Side: 0 Sector: 2

ADR :	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	-	*8*	*9*	*A*	*B*	*C*	*D*	*E*	*F*
0000:	F9	FF	FF	FF	FF	FF	FF	6F	-	00	07	80	00	09	F0	FF	0B
0010:	C0	00	0D	E0	00	FF	0F	01	-	11	20	01	13	F0	FF	00	00
0020:	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00

ROOT-DIRECTORY:

Drev:A Dos Sector: 15 Bios Spor: 0 Side: 1 Sector: 1

ADR :	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	-	*8*	*9*	*A*	*B*	*C*	*D*	*E*	*F*
0000:	46	49	4C	31	20	20	20	20	-	54	58	54	20	00	00	00	00
0010:	00	00	00	00	00	00	27	4F	-	AC	18	02	00	06	00	00	00
0020:	48	49	47	48	35	20	20	20	-	20	20	20	28	00	00	00	00
0030:	00	00	00	00	00	00	31	4F	-	AC	18	00	00	00	00	00	00
0040:	4B	41	54	31	20	20	20	20	-	20	20	20	10	00	00	00	00
0050:	00	00	00	00	00	00	36	4F	-	AC	18	03	00	00	00	00	00
0060:	53	43	52	45	45	4E	54	58	-	30	30	31	20	00	00	00	00
0070:	00	00	00	00	00	00	0E	59	-	AC	18	05	00	02	08	00	00
0080:	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00
0090:	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00

DIRECTORY (underkatalog):

Drev:A Dos Sector: 30 Bios Spor: 1 Side: 0 Sector: 1

ADR :	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	-	*8*	*9*	*A*	*B*	*C*	*D*	*E*	*F*
0000:	2E	20	20	20	20	20	20	20	-	20	20	20	10	00	00	00	00
0010:	00	00	00	00	00	00	36	4F	-	AC	18	03	00	00	00	00	00
0020:	2E	2E	20	20	20	20	20	20	-	20	20	20	10	00	00	00	00
0030:	00	00	00	00	00	00	36	4F	-	AC	18	00	00	00	00	00	00
0040:	4B	41	54	31	46	49	4C	20	-	54	58	54	20	00	00	00	00
0050:	00	00	00	00	00	00	49	4F	-	AC	18	04	00	0C	00	00	00
0060:	53	43	52	45	45	4E	54	58	-	30	30	31	20	00	00	00	00
0070:	00	00	00	00	00	00	0E	59	-	AC	18	0A	00	02	08	00	00
0080:	53	43	52	45	45	4E	54	58	-	30	30	32	20	00	00	00	00
0090:	00	00	00	00	00	00	14	59	-	AC	18	0F	00	02	08	00	00
00A0:	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00
00B0:	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00

a) Fortolk informationerne i BOOT-sektoren.

b) Fortolk informationerne i ROOT-directory og underkataloget. Label, filer og kataloger.
For hver ønskes navn, størrelse, filatribute (readonly, hiden m.m) samt start pladsering (clusternr).

c) Find ved hjælp af FAT-tabellen ud af, hvor underkataloget samt første fil i dette er pladseret filerne er pladseret på disketten (clustre).

Opgave FIL008.

Nedenstående er DUMP af BOOT-sektor, FAT-tabel og ROOT-directory fra en noget speciel diskette:

DUMP AF BOOT-sektor (uddrag)

```
0000:  EB 40 90 46 4C 20 76 31 - 2E 32 34 00 02 04 01 00     δ@ÉFL v1.24.....
0010:  02 80 01 32 01 F9 02 00 - 06 00 01 00 00 00 00 00     .Ç.2.˙..........
0020:  00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00     ................
0030:  00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00     ................
0040:  00 00 8C C8 3D 00 7C 74 - 08 BE 73 02 E8 6A 00 CD     ..î╚=.│t.╛s.Φj.=
```

DUMP af FAT-tabel (uddrag) **12-BITS**

```
0000:  F9 FF FF 03 40 00 FF 6F - 00 07 A0 00 09 F0 FF 0B     ˙.@.o..á..≡.
0010:  C0 00 FF FF FF FF FF FF - FF 2F 01 13 40 01 FF FF     └./..@.
0020:  FF FF 8F 01 FF AF 01 1B - C0 01 1D E0 01 FF 0F 02     Å.¤..└..α...
0030:  21 F0 FF FF 4F 02 FF 6F - 02 FF 8F 02 FF AF 02 FF     !≡O.o.Å.¤.
0040:  CF 02 FF EF 02 FF 0F 03 - FF 2F 03 FF 4F 03 FF 6F     ╧.∩.../.O.o
0050:  03 FF 0F 00 00 00 00 00 - 00 00 00 00 00 00 00 00     ................
0060:  00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00     ................
```

Dump af ROOT-directory (uddrag)

```
0000:  53 50 45 43 44 49 53 4B - 20 20 20 28 00 00 00 00     SPECDISK   (....
0010:  00 00 00 00 00 00 AF 56 - B6 1A 00 00 00 00 00 00     ......¤V╢.......
0020:  46 31 20 20 20 20 20 20 - 46 49 4C 21 00 00 00 00     F1      FIL!....
0030:  00 00 00 00 00 00 4F 58 - B5 1A 05 00 20 2F 00 00     ......OX╡... /..
0040:  46 49 4C 32 20 20 20 20 - 54 53 54 20 00 00 00 00     FIL2    TST ....
0050:  00 00 00 00 00 00 FB 76 - A2 1A 02 00 00 18 00 00     ......√vó.......
0060:  46 49 4C 33 20 20 20 20 - 54 53 54 20 00 00 00 00     FIL3    TST ....
0070:  00 00 00 00 00 00 6D 79 - A2 1A 22 00 01 00 00 00     ......myó."·····
0080:  46 49 4C 34 20 20 20 20 - 54 53 54 20 00 00 00 00     FIL4    TST ....
0090:  00 00 00 00 00 00 3B 87 - A2 1A 08 00 00 10 00 00     ......;çó.......
00A0:  55 4B 41 54 20 20 20 20 - 20 20 20 10 00 00 00 00     UKAT        ....
00B0:  00 00 00 00 00 00 0B BB - BF 1A 0D 00 00 00 00 00     .......╗┐.......
```

a) Fortolk informationerne i BOOT-sektoren og angiv følgende i hex og decimat format:

- Antal bytes pr. sektor.
- Antal sektorer pr. cluster.
- Antal entries i ROOT.
- Antal sektorer ialt på medie.
- Antal sektorer pr. spor.
- Antal sider.

Udregn antal spor:
(sektorer ialt = antal sider * antal sektore pr. spor * antal spor)

b) Med udgangspunkt i ROOT-dumpet ønskes følgende beskrevet:

- Er SPECDISK en fil eller Hvad ? Svar skal begrundes.
- Hvilken dato og tidspunkt er filen F1.FIL fra ? (svar i klar tekst - decimalt)
- Hvad er F1.FIL's attributer ? (svar i klar tekst).
- Hvad er F1.FIL's størrelse ? (decimalt)

c) Med udgangspunkt i ROOT-dumpet og FAT-tabellen ønskes følgende:

- Angiv F1.FIL's pladsering på disketten (hvilke clustre optager den).
- Angiv hvor meget optager filen (F1.FIL) fysisk på disketten. (opgives i både antal clustre og antal sektorer.
- Angiv hvor meget spildplads er der for denne fil (intern fragmentering).

4.

Maskinsprog og assembleringsprocessen

Opgaver

Indhold:

4.1.a

Modelmaskinen
Opgaver

Bjørk Busch

Opgave 1.

Indlæg tre tal i celle 20, 21 og 22.
Udarbejd et program, som kan addere tallene i celle 20 og 21 og derefter subtrahere tallet i celle 22. Resultatet gemmes i celle 23.

Opgave 2.

Fremstil et program i symbolsk maskinsprog til model-maskinen (papirløsning), der udfører følgende:

celle 39 := celle 27 * celle 38

Progammet **må ikke** bruge instruktionen multiply, men skal løse opgaven ved brug af ADD + løkke.

Opgave 3.

Givet er følgende program i model-maskine maskinkode, hvor celle 00 indeholder første instruktion.

```
000:01000010
001:02000010
002:06000011
003:99000000
004:
005:
006:
007:
008:
009:
010:00000020
011:
```

Oversæt maskinkoden til symbolsk maskinsprog og forklar hvad programmet laver (evt. i pascal notation)

Opgave 4.

Udarbejd et program, der dividerer tallet i celle 21 med tallet i celle 22.
Resultatet gemmes i celle 23.
Prøv at se hvad der sker hvid tallet 0 lægges i celle 22.

Opgave 5.

Udbyg programmet fra foregående opgave, således at det nu undersøges inden divisionen, om tallet i celle 22 er nul. Hvis dette er tilfældet, skal divisionen ikke udføres; men der flyttes i stedet for værdien nul til celle 23.

Opgave 6.

Programmet fra foregående opgave udbygges, således at vi nu også finder resten ved divisionen. Resten gemmes i celle 24.

Opgave 7.

Givet er følgende program i model-maskine maskinkode (den decimale), hvor celle 00 indeholder første instruktion.

Cellenr	Indhold	Cellenr	Indhold
00:	01000009	10:	00000044
01:	02000010	11:	00004030
02:	06000013	12:	00003000
03:	01000011	13:	00000000
04:	03000012	14:	00000000
05:	05000013	15:	00000000
06:	06000008	16:	00000000
07:	99000000	17:	00000000
08:	00000000	18:	00000000
09:	00000012	19:	00000000

Oversæt maskinkoden til symbolsk maskinsprog og forklar hvad programmet laver (evt. i pascal notation)

Opgave 8.

I celle 25, 26 og 27 indlægges henholdsvis månedsløn, månedsfradrag og trækprocent, hvorefter skatteberegning foretages. Den beregnede A-skat gemmes i celle 28, og nettobeløbet gemmes i celle 29.

Opgave 9.

Givet er nedenstående program skrevet i symbolsk maskinsprog til model-maskinen.

```
Linie
  1  PGSTART        START
  2                 LOAD          A
  3                 SUBDTRACT     A
  4  BACK           TESTGREATER   K
  5                 JUMP          FORWARD
  6                 JUMP          NEXT
  7  NEXT           ADD           E
  8                 STORE         A
  9                 JUMP          BACK
 10  FORWARD        TESTZERO
 11                 JUMP          GOON
 12                 STOP
 13  GOON           SUBTRACT      E
 14                 STORE         A
 15                 JUMP          FORWARD
 16  E              DATA          1
 17  K              DATA          5
 18  A              RESERVE       1
 19                 END           PGSTART
```

a) Skrivebordstest ovenstående assemblerprogram.

b) Assembler herefter programmet (skrivebordsløsning) **med syboltabel.**

Opgave 10.

Fra celleadresse 35 og frem indlægges en tabel med 1 til 4 tal (1 i hver celle). Tabellen afsluttes med tallet -1 (99999999).
Udarbejd et program til modelmaskinen, der kan udregne et gennemsnit af tallene i tabellen og lagre dette i celle 34, idet tallet -1 (99999999) ikke skal medtages.

Opgave 11.

I celle 25 indlægges et beløb i øre.
Fremstil et program til modelmaskinen, der kan beregne moms i celle 26 og beløb incl. moms i celle 27, idet beløb incl. moms skal afrundes til hele 25-ører.

Opgave 12.

Givet er nedenstående program skrevet i symbolsk maskinsprog til modelmaskinen.

```
Linie
  1  PGSTART        START
  2                 LOAD          T1
  3                 TESTZERO
  4                 JUMP          DIV
  5                 LOAD          K1
  6                 STORE         T1
  7  DIV            LOAD          T2
  8                 DIVIDE        T1
  9                 STORE         T3
 10                 MULTIPLY      T1
 11                 STORE         H1
 12                 LOAD          T2
 13                 SUBTACT       T3
 14                 STORE         T4
 15  K1             DATA          1
 16  T1             DATA          3
 17  T2             DATA          25
 18  T3             RESERVE       1
 19  T4             RESERVE       1
 20  H1             RESERVE       1
 21                 END           PGSTART
```

a) Skrivebordstest ovenstående assemblerprogram.

b) Assembler herefter programmet (skrivebordsløsning) **med syboltabel.**

Opgave 13.

Fibonacci-tal er en talrække der starter med 1. Næste tal er 1+1=2. Næste tal er 1+2=3, næste er 2+3=5, næste er 3+5=8 .. 5+8=14 .. 8+14=22 o.s.v.
Skriv et program der genererer så mange fibonacci-tal som muligt i lageret. Husk her på følgende lille finesse **Selvmodificerende programmer kan overskrive sig selv!**

Opgave 14.

Givet er nedenstående program skrevet i symbolsk maskinsprog til modelmaskinen.

```
PGSTART    START
           LOAD       BELOB
           TESTZERO
           JUMP       BEREGN
           JUMP       SLUT
BEREGN     MULTIPLY   MOMSPRO
           DIVIDE     HUNDRED
           STORE      MOMSBEL
           ADD        BELOB
           STORE      BELOB
SLUT       STOP
MOMSPRO    DATA       22
HUNDRED    DATA       100
BELOB      DATA       3210
MOMSBEL    RESERVE    1
           END        PGSTART
```

Assembler programmet og vis hvordan symboltabelen kommer til at se ud.

Opgave 15.

Skrivebordstest programmet fra foregående opgave, idet du viser indhold i akkumulatorregister, adresseregister, programtæller samt relevante dataceller efter udførslen af hver maskininstruktion.
Gennemfør herefter skrivebordstesten, idet det denne gang forudsættes at feltet BELOB indeholdt værdien 0.

Opgave 16.

Du bedes udarbejde et program i symbolsk maskinsprog til modelmaskinen, der kan udregne summen af tallene i en tabel.
Tabellen pladseres fra celleadresse 37 og nedefter (så mange der nu kan blive plads til).
I celleadresse 38 indlægges antallet af elementer i tabellen.
Den beregnede sum pladseres i celleadresse 39.

Skitse til løsning i "PASCAL" notation.

```
Sum := 0;
Celleadr := 37;
While AntalElementer <> 0 do begin
     Sum := Sum + Celle[Celleadr];
     Celleadr := Celleadr - 1;
     AntalElementer := AntalElementer - 1;
End;
```

5.

Assemblermaskinen

Opgaver

Indhold:

5.1.b

DOS environment blok
Opgaver

Bjørk Busch

Opgave 1.

Nedenstående vise et dump fra en PC's environment-blok, hvor der opbevares informationer om PROMPT, PATH m.m.
Indholdet kan udskrives med SET kommandoen i DOS.

Blokken starter i begyndelsen af dumpet.
De enkelte parametre er adskilt af en byte med binære nuller.
Sidste parameter i blokken efterfølges af en ekstra byte med binære nuller.

De enkelte parametre opbevares i ASCII format.

ADR : Seg : Ofs	0	1	2	3	4	5	6	7		8	9	A	B	C	D	E	F
1050:0000	44	4F	53	3D	45	3A	5C	4D	-	53	44	4F	53	35	30	00	50
1050:0010	52	4F	4D	50	54	3D	24	65	-	5B	34	31	3B	33	30	6D	24
1050:0020	50	24	47	24	65	5B	34	33	-	3B	33	30	6D	00	43	4F	4D
1050:0030	53	50	45	43	3D	45	3A	5C	-	4D	53	44	4F	53	35	30	5C
1050:0040	43	4F	4D	4D	41	4E	44	2E	-	43	4F	4D	00	50	41	54	48
1050:0050	3D	45	3A	5C	41	53	4D	3B	-	5C	3B	45	3A	5C	42	41	54
1050:0060	46	49	4C	45	52	3B	45	3A	-	5C	55	54	49	4C	49	54	59
1050:0070	3B	45	3A	5C	4D	53	44	4F	-	53	35	30	00	00	01	00	45
1050:0080	3A	5C	4D	53	44	4F	53	35	-	30	5C	44	45	42	55	47	2E
1050:0090	45	58	45	00	8F	4E	44	2E	-	41	53	4D	45	44	00	54	48
1050:00A0	5A	65	10	9A	8F	9A	F0	FE	-	44	45	42	55	47	00	4B	01
1050:00B0	CD	20	FF	9F	00	9A	F0	FE	-	1D	F0	DC	01	3A	0F	4B	01
1050:00C0	3A	0F	56	01	3A	0F	3A	0F	-	01	03	01	00	02	FF	FF	FF
1050:00D0	FF	FF	FF	FF	FF	FF	FF	FF	-	FF	FF	FF	FF	5A	10	B9	49
1050:00E0	76	10	14	00	18	00	65	10	-	FF	FF	FF	FF	00	00	00	00
1050:0090	05	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00

4.1) Hvad er den aktuelle PROMPT ?

4.2) Hvad står der i den sidste parameter ?

5.2.a

8086 assemblerprogrammer
Opgaver

Bjørk Busch

Opgave 1.

Omsæt følgende programstump i pascal-notation til 8086 assembler:

Result := (Tal1 + Tal2) * 3

Alle variable er af typen integer. Find selv på testværdier.
Multiplikation skal klares uden brug af multiplikations-instruktioner og uden løkke (konstant 3 gange).

Opgave 2.

Udbyg ovenstående programstump med følgende (efter beregning i opgave 1):

```
IF (Result < 0) then
     Result := Result + 100
ELSE (IF Result = 0) then
     Result := 20
ELSE
     Result := Result - 100;
```

Opgave 3.

Omsæt følgende programstump i pascal-notation til 8086 assembler:

```
SUM := 0;
FOR i := 1 to 4 DO BEGIN
     SUM := SUM + ITAB[i];
END;
```

Tabellen kan defineres således i assembler:

```
ITAB        DW          11,21,31,41
```

TIPS! Der skal anvendes base-/index-register som ved eksempel fra MOV. Denne adressering kan også anvendes i forbindelse med ADD.

Opgave 4.

Givet er følgende datafelter:

```
INAVN           DB    'EDB-SKOLEN','$'
UNAVN           DB    30 DUP (' ')
                DB    13,10,'$'
```

Det er med vilje, at ascii-tegnene 13, 10 og $ kommer efter UNAVN og at der ikke påhæftes et symbolsk navn (symbolsk-adresse).

Du bedes udarbejde en assembler-rutine, der kan flytte teksten (excl. $) fra INAVN til UNAVN, idet der skal være et **mellemrum på et tegn** mellem hvert bogstav i UNAVN.

Opgave 5.

Skriv programmet fra opgave 4 så det kan oversættes med MASM, LINKES og aftestes.

Prøv af afvikle programmet under CODEVIEW.

Opgave 6.

Programmet fra opgave 5 skal nu udbygges, så de to tekster udskrives på skærmen (DOS funktionskald 9 - int 21h).

Offset-adressen på en variabel kan fås på følgende måde:

```
MOV  DX,Offset xxxxx

Hvor xxxxx er det symbolske navn.
```

Prøv programmet under CODEVIEW, samt direkte ude i DOS.

Opgave 7.

Lav en rutine, der kan optælle antallet af ascii-tegnet A i en tekstvariabel, idet variablen afsluttes med et binært nul.

Opgave 8.

Lav en procedure, der kan omsætte et binært tal (16 bit word) til ascii-format. Proceduren skal hedde **BIN2ASC** og være at typen **NEAR**.

Proceduren skal kaldes med følgende parametre overført i registre:

Ax	Word med det binære tal, der skal konverteres.
Cx	Antallet af cifre, der skal være i det konverterede tal (mindst 1 ciffer).
Ds:Di	Startadressen på en tekststreng med plads til tallet i asciiformat.

Eksempler på hvordan proceduren skal kunne bruges:

```
        MOV     Ax,BinTal1
        LEA     Di,AscTal1
        MOV     Cx,2
        CALL    BIN2ASC
        LEA     Dx,Linie1
        CALL    UDSKRIV

        MOV     Ax,BinTal2
        LEA     Di,AscTal2
        MOV     Cx,3
        CALL    BIN2ASC
        LEA     Dx,Linie2
        CALL    UDSKRIV
```

Rutinen UDSKRIV er defineret på følgende måde:

```
UDSKRIV PROC    NEAR
        PUSH    Ax          ;Gem Ax på STAK
        MOV     Ah,09h      ;Kald DOS for
        INT     21          ;    udskrift
        POP     Ax          ;Restore Ax fra STAK
        RET                 ;Returer
UDSKRIV ENDP
```

I datasegmentet er følgende definitioner:

```
BinTal1 Dw      27
Linie1  Db      'Første tal var: '
AscTal1 Db      2 DUP (?)
        Db      10,13,'$'       ;Slut med linieskift
BinTal2 Dw      743
Linie2  Db      'Andet tal var: '
AscTal2 Db      3 DUP (?)
        Db      10,13,'$'       ;Slut med linieskift
```

Opgave 8 (fortsat).

Hjælp til program-algoritme:

- Start med at gemme alle registre, der ændres i proceduren, på stakken.
- Sæt Di til at pege på det sidste ciffer i tallet (det mindst betydende - højre).
- Ved heltalsdivision med 10 fås nu en kvotient og en rest. 123/10 giver 12 som kvotient og 3 som rest.
- Resten som jo er et binært tal mellem 0 og 9 vil kunne stå i en byte og kan dirrekte omregnes til et ascii-tegn.
  Find værdien i ascci-tabellen.
- Det beregnede ascii-tegn flyttes nu på plads i lageret.

  Ved at gentage division med 10 og det efterfølgende fås et ciffer fra højre af gangen, og dette skal så ske et antal gange svarede til hvad proceduren fik leveret i Cx.
- Slut af med at restore alle de gemte registre fra stakken og returner til kaldende rutine.

Opgave 9.

Lav en procedure, der kan omsætte et tal i ascii-format til binært (16 bit word).

Proceduren skal hedde **ASC2BIN** og være at typen **NEAR**.

Proceduren skal kaldes med følgende parametre overført i registre:

Ds:Si	Startadressen på en tekststreng med tallet i ascii-format.
Cx	Antallet af cifre, der skal konverteres.
Ax	Resultatet (det konverterede tal).

Hjælp til program-algoritme:

```
BINTAL := 0;
FOR I:= 1 To ANTALCIFRE DO BEGIN
     BINTAL := BINTAL * 10;
     BINTAL := BINTAL + ( ORD(ASCTAL[I]) - ORD('0') );
END;
```

Opgave 10.

```
DATA-SEGMENT
ADRESSE     HEX-DUMP AF DATASEGMENT
seg :ofs                  FØR KØRSEL
2026:0000   05 00 0B 00 16 00 21 00-2C 00 2D 00 00 00
seg :ofs                  EFTER KØRSEL
2026:0000   05 00 0B 00 16 00 21 00-2C 00 2D 00 9B 00
            ----- ----- ----- ----- ----- ----- -----
            Antal -------------Itab------------ SUM
```

```
CODE-SEGMENT
ADRESSE    MASKINKODE    SYMBOLSK KODE          kommentarer
seg :ofs                                        (ikke DEBUG)
2027:0000 E80600         CALL  0009             NB 0006 relativ
2027:0003 E80F00         CALL  0015             NB 000F relativ
2027:0006 E80600         CALL  000F             NB 0006 relativ

2027:0009 B82620         MOV   AX,2026
2027:000C 8ED8           MOV   DS,AX
2027:000E C3             RET

2027:000F B000           MOV   AL,00
2027:0011 B44C           MOV   AH,4C
2027:0013 CD21           INT   21

2027:0015 33C0           XOR   AX,AX
2027:0017 33F0           XOR   SI,SI
2027:0019 8B0E0000       MOV   CX,[0000]
2027:001D 83F900         CMP   CX,+00
2027:0020 7E0A           JLE   002C             NB 0A rel. +10
2027:0022 03840200       ADD   AX,[SI+0002]
2027:0026 83C602         ADD   SI,+02
2027:0029 49             DEC   CX
2027:002A EBF4           JMP   0020             NB F4 rel. -12
2027:002C A30C00         MOV   [000C],AX
2027:002F C3             RET
```

a) Ovenstående viser et assemblerprogram udskrevet i DEBUG-format.
Foretag sammenligning med programmet, som det så ud i kildetekst (på næste side).

b) Foretag en skrivebordstest af programmet, hvor relevante registre og lagerceller vises efter udførslen af hver enkelt instruktion.

c) Foretag en aftestning af programmet med brug af CODEVIEW, idet der køres "single-step", så der kan sammenlignes med skrivebordstest.
Programmet kan udleveres på diskette og hedder ASMPROC1

Opgave 11.

Udbyg det udleverede program med en rutine, der kan konvertere den binære sum til ASCII format, samt en rutine der kan udskrive den konverterede sum på skærmen.

```
DATA    Segment Para Public 'DATA'
Antal   Dw      5                       ; Antal el. i ITab
ITab    Dw      11,22,33,44,45
Sum     Dw      ?
DATA    EndS                            ; slut på data-segment


CODE    Segment Para Public 'CODE'
        Assume Cs:CODE,Ds:DATA,Es:nothing,Ss:STACK

; Proceduren Main er den rutine der får kontrollen ved start
Main    Proc    Near
        Call    InitDS                  ; udfør InitDS
        Call    Beregn                  ; udfør Beregn
        Call    StopPG                  ; udfør StopPG
Main    EndP

; Proceduren Opstart opsætter adresse på datasegmentet
InitDS  Proc    Near
        Mov     Ax,Seg DATA             ; etabler
        Mov     Ds,Ax                   ;     datasegment
        Ret                             ; retur fra procedure
InitDS  EndP

; Proceduren Afslut overgiver kontrollen til DOS igen
StopPG  Proc    Near
        Mov     Al,0                    ; returkode = 0
        Mov     Ah,4Ch                  ; opsæt funktion for
        Int     21h                     ; terminate og udfør
;  Da denne funktion ikke giver kontrollen tilbage
;  er der ingen Ret-instruktion
StopPG  EndP

; Proceduren Beregn udregner summen af tal i ITab
Beregn  Proc    Near
        Xor     Ax,Ax                   ; Ax := 0
        Xor     Si,Si                   ; Si -> første elm.
        Mov     Cx,Antal                ; FOR Cx := Antal
        Cmp     Cx,0                    ;  DOWNTO 1
For:    Jle     EndFor                  ;   2. runde Dec Cx
        Add     Ax,ITab[Si]             ;   Ax := Ax+ITab.elm.
        Add     Si,2                    ;   Si -> næste elm.
        Dec     Cx                      ; næste cx i for-løkke
        Jmp     For
EndFor:                                 ; END
        Mov     Sum,Ax                  ; Sum := Ax
        Ret                             ; retur fra procedure
Beregn  EndP

CODE    EndS                            ; slut på code-segment


STACK   Segment Para Stack 'STACK'
        dw      128 dup(?)              ; 128 ord afsat
STACK   EndS                            ; slut på stack-seg


        End     Main                    ; Slut og opsæt start
```

Opgave 12.

```
Upper    Proc    Near
         Push    Ax                ; save arbejdsregister
         Push    Bx                ; save arbejdsregister
         Mov     Bx,Dx             ; Bx := adr. index
Loop1:
         Mov     Al,[Bx]           ; While "bogstav"
         Cmp     Al,'$'            ;    Ds:[Bx] <> '$'
         Je      EndLoop1          ;       Do
;        ......
;        ......
;        ......
         Inc     Bx                ;  Ds:Bx => næste bog.
         Jmp     Loop1             ; EndWhile
EndLoop1:
         Pop     Bx                ; restore register
         Pop     Ax                ; restore register
         Ret                       ; returner
Upper    EndP
```

Ovenstående er udgangspunkt for en rutine, der skal kunne rette alle bogstaver i en tekst til store bogstaver.
Ved kald af rutinen indeholder Ds:Dx adressen på første tegn i teksten. Teksten afsluttes med et $-tegn.
Du bedes gøre rutinen færdig.
Rutinen kan aftestes ved skrivebordstest, eller den kan aftestes maskinelt. Der kan udleveres et program (ASMUPPER), der indeholder den påbegyndte rutine samt testkald.

Opgave 13.

Givet er følgende programstump:

```
0
1            Mov     Bx,6
2            Mov     Ax,1
3    L1:
4            Mov     DataA[Bx],Al
5            Shl     Ax,1            ; bits 1 plads left
6            Dec     Bx
7            Jnl     L1
8
```

DataA er defineret på følgende måde:

```
DataA    Db       16 Dup (0)
```

Skrivebordstest programstumpen.
Hvad indeholder DataA og de berørte registre (Ax og Bx) efter rutinen er kørt.

Opgave 14.

Givet er følgende definitioner i datasegmentet:

```
A       Db      '00123'

B       Db      '45678'

C       Db      6 dup(?)
```

Samt følgende programstump:

```
Linie          Programkode
01    P1:
02             Mov     Bx,4
03             Mov     Al,0
04             Mov     Dl,10
05             Mov     Dh,(48*2)+10    ; udregnes af
06                                     ; oversætter
07             Mov     Cl,8
08    L1:
09             Shl     Ax,Cl
10             Add     Ah,A[Bx]
11             Add     Ah,B[Bx]
12             Sub     Ah,Dh
13             Jl      L2
14             Inc     Al
15             Jmp     L3
16    L2:
17             Add     Ah,Dl
18    L3:
19             Or      Ah,48
20             Mov     C+1[Bx],Ah
21             Dec     Bx
22             Jnl     L1
23
24             Or      Al,48
25             Mov     C,Al
26
27             Ret
```

a) Foretag en skrivebordstest af ovenstående programstump, idet alle relevante data (herunder registre) vises efter udførsel af de enkelte programlinier.

b) Udarbejd en beskrivelse af ovenstående programstump.

Opgave 15.

Fremstil et assemblerprogram, der kan ændre farven på alle tegn på skærmen til grøn tekst på rød baggrund.
Programmet skal anvende direkte acces til skærmbufferen.
Skærmen kører i alm. tekstmode med 80 tegn pr. linie og 25 linier (2000 tegn).
Skærmbufferen er pladseret i adresse B800:0000

løsning i PASCAL.

```
Program NyATT;
VAR
    i,SkrmSeg,SkrmOfs: Integer;
BEGIN
     SkrmSeg := $B800;
     SkrmOfs := $0000;                          { b BAG i FOR }
     FOR i := 1 to 2000 DO BEGIN                { l RGB n RGB }
          MEM[SkrmSeg:SkrmOfs+1] := $42;        { 0 100 0 010 }
          SkrmOfs := SkrmOfs + 2;
     END;
END.
```

Opgave 16.

Nedenstående viser en programstump i PASCAL til søgning i en usorteret tabel.

```
VAR
     SoegeTal: Integer;
     AntalElm: Integer;
     i:        Integer;
     Fundet:   Boolean;
     Tab:      Array[1..200] of Integer;
............
............
i := 0;
Fundet := False;
WHILE (NOT Fundet) AND (i < AntalElm) DO BEGIN
     i := i + 1;
     IF (SoegeTal = Tab[i]) THEN
          Fundet := True;
END;

IF Fundet then
     WRITELN('Fundet')
ELSE
     WRITELN('Ikke Fundet');
```

Omskriv ovenstående programstump til 8086 assembler.
Når programstumpen kører er AntalElm sat til 200 og tabellen er blevet fyldt op.

Opgave 17.

Fremstil et assemblerprogram, der kan indlæse en dato fra tastaturet, beregne det tilhørende dagnr og udskrive dette på skærmen.

Opgave 18.

Fremstil et assemblerprogram, der kan hente dagsdato fra systemet med DOS-kald og udskrive månedens navn på skærmen.
(Januar, Februar ..... December)

Udbyg evt. programmet så også dag og årstal udskrives.
(eksempel. 10. Januar 1993)

Opgave 19.

Fremstil et assemblerprogram, der kan sammenligne første linie fra en "fil" (standard input) med en tekst, der angives som parameter til programmet. Hvis der er forskel afleveres en returkode > 0 til DOS ellers retureres med 0.

Parameteren fås fra PROGRAM-SEGMENT-PREFIX'et (PSP), hvor der er afsat plads til en streng på 127 tegn. Længden af strengen er angivet i en byte lige før strengen (svarende til formatet i en STRING i PASCAL).
Ved opstart af såvel EXE- som COM-programmer vil Es og Ds indeholde segmentadressen på PSP'et.

Filen kan redirigeres eller pipes ind i programmet, således at dette kan læse fra standard input (som fra tastatureu).
DOS-funktion 0A kan anvendes såvel som 3F.
Den sidste funktion arbejder med et såkaldt HANDLE-nummer, der dels kan tildeles fra DOS når en fil åbnes (med en anden DOS-funktion 3D), dels knyttes til standardenhederne:

Handle'nr:	Standardenhed
0	Standard input (keyboard, redir, pipe)
1	Standard output (Skærm, redir, pipe)
2	Standard error (skærm kan ikke undertrykkes med redir eller pipe)

Opgave 20.

Fremstil et program, der kan finde DOS'ens environment blok, hvor PROMPT, PATH m.m. findes og herefter udskrive indholdet af PATH parameteren.
Segmentadressen på environmentblokken kan findes via PSP'et, hvor adressen opbevares.

Opgave 21.

Fremstil et program (WRITEENV), der kan udskrive en parameter fra environment-blokken.
Navnet på parameteren angives som parameter til programmet ved opstarten.
Eks. WRITEENV PATH
Dette skulle give en udskrift af PATH-parameteren

Opgave 22.

Fremstil en assembler rutine, der kan rykke alle linier på skærmen en linie ned (SCROLLE en linie baglæns).
Der skal anvendes BIOS-kald (INT 10)

Opgave 23.

Fremstil en assembler rutine, der kan slette en firkant midt på skærmen med øverste venstre hjørne i linie 10, kolonne 30 og nederste højre hjørne i linie 16, kolonne 60.
Der skal anvendes BIOS-kald (INT 10)

Opgave 24.

Fremstil en assembler rutine, der kan udskrive en tekst på skærmen ved brug af BIOS-kald.

Opgave 25.

Fremstil en assembler rutine, der kan indlæse et password fra tastaturen uden at det udskrives på skærmen.

Opgave 26.

Fremstil en assembler rutine, der kan teste om venstre eller højre shift-tast er trykket ned.

Opgave 27.

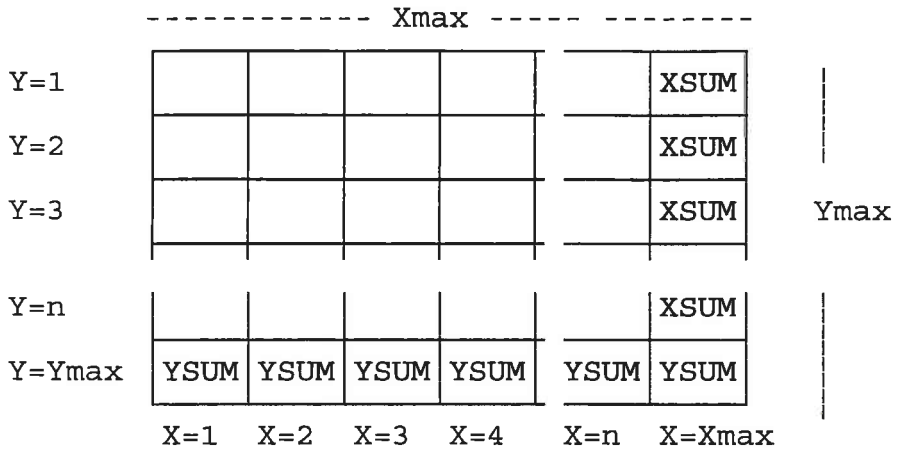
Fremstil et assembler program, der kan indlæse en ASCII-fil og udskrive en indholdet i en ny fil, idet alle bogstaver konverteres til store (anvend redirektion for input og output).

Opgave 28.

Fremstil en rutine, der kan udskrive en byteværdi binært (8 ascii 0/1-taller)

Opgave 29.

2-dimmensionel tabel:



Ovenstående viser en 2-dimensionel tabel af integer.
Tabellen er udlagt i lageret således at elementerne første række (Y) kommer efter hinanden, efter sidste felt i rækkken (sumfelt) følger første felt fra næste række. Felterne i samme søjle (X) ligger således ikke ved siden af hinanden, men med en afstand svarende til længden af en række.

Xmax: Angiver antal elementer i en række (antal søjler) incl. sumfelt.
Ymax: Angiver antal elementer i en søjle (antal rækker) incl. sumfelt.

a) Udarbejd en assemblerrutine, der kan beregne summen af hver række (XSUM) excl. sidste (Y=Ymax).

Løsningsskitse i PASCAL:

```
FOR Y := 1 to Ymax-1 DO BEGIN
      SUM := 0;
      FOR X := 1 to Xmax-1 DO BEGIN
            SUM := SUM + TAB[Y,X];
      END;
      TAB[Y,X] := SUM;
END;
```

b) Udarbejd en assemblerrutine, der kan beregne summen af hver søjle incl. den sidste (X=Xmax).

Løsningenskitse i Pascal ligner ovenstående temmelig meget, så derfor gives her nogle andre tips.

Da der skal springes tilbage i adresserne ved start på hver søjle, er det en ide, at fastholde adressen på første elm. i den søjle der behandles. Starten på næste søjle kan nemt bereges da den er et elm. højere. Brug BP registeret til dette formål. Brug nu SI registret til at holde styr på den aktuelle element i søjlen, idet BP angiver start på søjle og SI afstand herfra. Afstanden mellem elementerne i søjlen, kan beregnes som XMAX*elm.længde.

Opgave 30.

Udarbejd et assemblerprogram **PARCOUNT**, der kan optælle antal parametre i parameterlisten, og returnere antallet som ERROR-CODE, der kan aftestes i DOS som IF ERRORLEVEL.

Opgave 31.

Udarbejd et assemblerprogram **TRANSLATE**, der kan anvendes som filter i DOS (læse std. input og skrive std. output).
Programmet skal omsætte alle bogstaver til **store**.

Opgave 32.

Udbyg foregående program, så der kan angives parameteren /L, hvorved alle bogstaver skal omsættes til små istedet for store. Hvis parameteren ikke er sat skal der stadig omsættes til store bogstaver.

Opgave 33.

Udbyg foregående program, så der kan angives parameteren /A, hvorved alle tegn mindre end blanke (32), med undtagelse af CR, LF og FF, samt alle tegn større end 127 skal konverteres til blanke.

Opgave 34.

```
01: P2      PROC    NEAR
02:         Mov     Dx,Cx
03:         Mov     Cl,4
04:         Xor     Ah,Ah
05:         Xor     Bh,Bh
06: P2L1:
07:         Xor     Di,Di
08:         Mov     Ch,8
09: P2L2:
10:         Mov     Al,[Si]
11:         Shl     Ax,Cl
12:         Shr     Al,Cl
13: R1:
14:         And     Ah,00001111B
15:         Mov     Bl,Ah
16:         Mov     Ah,DataC[Bx]
17:         Mov     DataB[Di],Ah
18: R2:
19:         Inc     Di
20:         Mov     Bl,Al
21:         Mov     Al,DataC[Bx]
22:         Mov     DataB[Di],Al
23:         Inc     Di
24:         Inc     Di
25:         Inc     Si
26:         Dec     Ch
27: R3:
28:         Jnz     P2L2
29:         Mov     Bx,Dx
30:         Mov     Dx,Offset DataB
31:         Mov     Ah,09h
32:         Int     21h
33:         Mov     Dx,Bx
34:         Cmp     Si,Dx
35:         Jl      P2L1
36:         Ret
37: P2      ENDP
```

Datasegment indeholder:

```
DataA   Db      128 Dup (0)     ; 128 byte med binær 0
DataB   Db      '.. .. .. .. .. .. .. ..',13,10,'$'
DataC   Db      '0123456789ABCDEF'
```

a) Udfør en skrivebordstest af et gennemløb af assemblerrutinen P2, når følgende oplyses:
- DataA har fået nyt indhold, nemlig 4 ASCII-tegn "TEST" efterfulgt af 124 binære nuller.
- Cx indeholder ved starten af P2 værdien 4.
- Si indeholder offset på DataA, her 0.

Skrivebordstesten skal dokumenteres, ved at notere indholdet af registre og relevante lagerceller på følgende steder i programmet P2L1, P2L2, R1, R2, R3 samt ved afslutningen af P2.

b) Forklar hvad rutinen laver.

Opgave 35.

Udarbejd et assemblerprogram "LIXTAL" i 8086 macro-assembler.
Programmet skal kunne optælle antallet af ord og bogstaver/tal i en asciitekst. Desuden udreges gennemsnitsordlængden som antal bogstaver divideret med antal ord.
Der er tale om et meget forsimplet lix-tal.

Programmet skal udskrive de tre nøgletal med passende ledetekst på skærmen.

Karakterer definers her som bogstaverne a-z,æ,ø,å,A-Z,Æ,Ø,Å samt tallene 0-9.

Et ord defineres som sammenhængende bogstaver/tal. Tegn der ikke er bogstaver/tal vil således adskille ord.

Teksten indlæses fra standard input, der kan være tastaturet eller en redirigeret fil.

Der kan indlæses med dos-funktion 0Ah, hvor der indlæses en linie af gangen, idet der så skal optræde en stopværdi til sidst (f.eks en tom linie eller et specialtegn).

Der kan også indlæses med dos-funktion 3Fh, hvor bufferen sættes stor nok til at kunne rumme hele teksten, der så kan indlæses på een gang.

Opgave 36.

Udarbejd et assemblerprogram "OPTAEL" i 8086 macro-assembler, som kan foretage en optælling af forekomsten af de enkelte ASCII-tegn (extended) i en tekstfil, der overføres som standard-input evt. pipe eller redirektion.

Eksempel BREV.TXT som input:

OPTAEL < BREV.TXT

Optællingen skal kun omfatte tegnene i intervallet 20 (hex) til AF (hex).

Resultalet af optællingen skal udskrives på skærmen.

Tips ! Tips ! Tips ! Tips !

Opbyg programmet i mindst 4 procedurer:

1. Styring.
2. Indlæsning i buffer.
3. Optælling for indlæst buffer.
4. Udskrift af slutresultat.

Vedrørende indlæsning fra DOS standardinput henvises til dos interrupt beskrivelser.

Indexer et tegn af gangen fra input-buffer (f.eks med SI registeret).
Definer evt. inputbuffer, så den kan rumme helle filen (sæt max til 4096).

Man kan evt. istedet indlæse eet tegn af gangen og stoppe når dette er CTRL-Z (ascii tegn nr. 26).

Optælling kan foretages på to måder:

1. Den pladsforbrugende men hurtige.

   Der oprettes en tabel på 256 ord initieret med nuller.
   Brug det indlæste tegn direkte som index til optællingstabel ved at gange værdien med 2 (længden af et element) (brug shift instruktion), tæl tabel-element som nu kan indexeres op med 1.
   Husk at ascii-tegnet er på 1 byte og index er på 2 byte, så vælg index-register med omhu.

2. Den pladsbesparende men langsommere.

   Der oprettes en tabel på 144 ord initieret med nuller (svarende til intervallet fra 20 hex til AF hex).
   Der subtraheres 20 (hex) fra det indlæste tegn, hvis resultatet er negativt hoppes uden om optælling.
   Der testes herefter om tegn er > 144 (AF-20 hex), hvis dette er tilfældet hoppes ligeledes uden om optælling.
   Den nye værdi anvendes herefter som index, idet der som før først ganges med 2.
   Fortsættes næste side.

Udskriften kan foretages på følgende måde:

Udskriv en linie for hvert af de aktuelle ascii-tegn (et af gangen).

For hvert element i optællingstabellen foretages en konvertering fra binært til ascii. Der henvises til opgave med konvertering fra binær til ascii.

Det enkelste er nok at optælle ascii-tegn og index til tabel hver for sig. Ascii-tegnen kan gemmes sammen med den "streng" der anvendes til konvertering, således at de kan udskrives på en gang.

Husk, hvis optællingsmodel 1 anvendt, så skal index starte 20 (hex) * 2 inde i tabellen. Der kan dog også angives et øget offset på tabellen.

Programmet kan **evt.** udbygges, så der ikke udskrives noget for uanvendte ascii-tegn.

For de meget entusiastiske kan programmet udbygges, således at der foretages en optælling af ordlængder (regn med max 20 tegn).
Et ord kan her defineres som afsluttet med en blank, et komma, et punktum samt alle de ascii-tegn, der ikke medtages i optællingen. Et ord starter med ethvert af de øvrige ascii-tegn.
Ved udbygning så start under alle omstændigheder med mindstekravet og udbyg så evt. senere.

6.

Pascalmaskinen

Opgaver

Indhold: