Program1:

```
DATA SEGMENT
string db 'HELLO WORLD!', '$'
DATA ENDS
CODE SEGMENT
ASSUME CS: CODE, DS: DATA
START
MOV AX, DATA
MOV DS, AX
LEA DX, string
MOV AH, 09H
INT 21H
MOV AH, 4CH
INT 21H
CODE ENDS
END START
```

Output:

edit msg.asm
masm msg.asm
msg.exe

Program2:

```
.MODEL SMALL
.STACK 100
.DATA
MSG DB "Hello World - My first 8086 Program !!!
$"
.CODE
MOV AX,@DATA
MOV DS,AX
LEA DX, MSG
MOV AH,09H
INT 21H
MOV AH,4CH
INT 21H
END
```

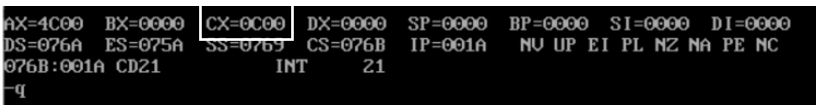
Output:

edit hello.asm
masm hello.asm
link hello.obj
hello.exe

```
DATA SEGMENT
N1 DW 0a00H
N2 DW 0200H
N3 DW ?
DATA ENDS
CODE SEGMENT
ASSUME CS :CODE;DS:DATA
START:
MOV AX,DATA
MOV DS,AX
XOR AX,AX
MOV BX,AX
MOV AX,N1
ADD AX,N2
MOV N3,AX
JNC STOP
INC BX
STOP:
MOV CX,AX
MOV AH,4CH
INT 21H
CODE ENDS
END START
```

Output 1:

```
edit addnum.asm
masm addnum.asm
link addnum.obj
debug addnum.exe
-t(until instruction INT 21H)
```



Program1:

```
DATA SEGMENT
N1 DW 2222H
N2 DW 0EEEEH
N3 DW ?
DATA ENDS
CODE SEGMENT
ASSUME CS :CODE;DS:DATA
START:
MOV AX,DATA
MOV DS,AX
XOR AX,AX
MOV BX,AX
MOV AX,N1
SUB AX,N2
MOV N3,AX
JNC STOP
INC BX
NOT AX
ADD AX, 0001H
STOP:
MOV CX,AX
MOV AH,4CH
INT 21H
CODE ENDS
END START
```

Program2:

```
CODE SEGMENT
ASSUME CS:CODE
START:
MOV AX,0000H
MOV BX,AX
MOV DX,AX
MOV SI, 3000H
MOV AX,[SI]
INC SI
INC SI
MOV BX,[SI]
INC SI
INC SI
SUB AX,BX
MOV [SI],AX
JC L1
INC SI
INC SI
MOV [SI],DX
INT 3
L1: INC DX
INC SI
INC SI
MOV [SI],DX
INT 3
CODE ENDS
END START
```

Output – Program1:

gedit sub-16bit1.asm
masm sub-16bit1.asm
link sub-16bit1.obj
debug sub-16bit1.exe

```
AX=4CCC  BX=0001  CX=CCCC  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=001F   NV UP EI NG NZ NA PE NC
076B:001F CD21           INT     21
```

Output – Program2:

gedit sub-16bit.asm
masm sub-16bit.asm
link sub-16bit.obj
debug sub-16bit.exe

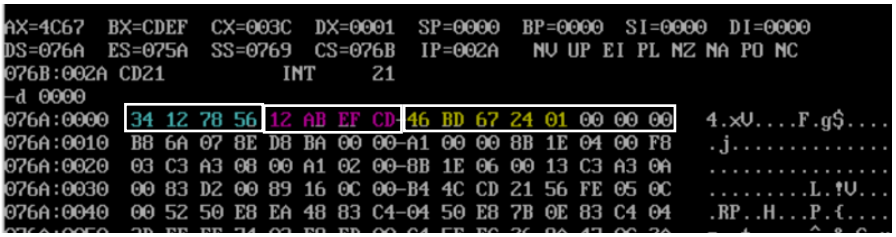
```
C:\>debug sub-16bit.exe
-g=0000

AX=9EDD  BX=5011  CX=0023  DX=0000  SP=0000  BP=0000  SI=3006  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=001C   NV UP EI PL NZ NA PE NC
076A:001C CC            INT     3
-e 3000
075A:3000  EE.ee   EE.ff   11.22   50.11

-g=0000

AX=EECC  BX=1122  CX=0023  DX=0000  SP=0000  BP=0000  SI=3006  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=001C   NV UP EI PL NZ NA PE NC
076A:001C CC            INT     3

-d 3004,3007
075A:3000            CC EE 00 00                             ....
-
```

Program :

```
DATA SEGMENT
NUM1 DW 1234H, 5678H
NUM2 DW AB12H, CDEFH
RESULT DW 3 DUP(?)
DATA ENDS
CODE SEGMENT
ASSUME CS:CODE, DS:DATA
MAIN:MOV AX,DATA
MOV DS,AX
MOV DX,00H
MOV AX,NUM1
MOV BX,NUM2
CLC
ADD AX,BX
MOV RESULT,AX
MOV AX,NUM1+2
MOV BX, NUM2+2
ADC AX,BX
MOV RESULT+2,AX
ADC DX,00H
MOV RESULT+4,DX
MOV AH,4CH
INT 21H
CODE ENDS
END MAIN
```

Output:

edit 32add.asm
masm 32add.asm
link 32add.obj
debug 32add.exe
-t(until INT 21H)



Program 1:

```
DATA SEGMENT
NUM1 DW 0BBBBH, 0EEEEH
NUM2 DW 1111H, 2222H
RESULT DW 3 DUP(?)
DATA ENDS
CODE SEGMENT
ASSUME CS:CODE, DS:DATA
MAIN:MOV AX,DATA
MOV DS,AX
MOV DX,00H
MOV AX,NUM1
MOV BX,NUM2
CLC
SUB AX,BX
MOV RESULT,AX
MOV AX,NUM1+2
MOV BX, NUM2+2
SBB AX,BX
MOV RESULT+2,AX
SBB DX,00H
MOV RESULT+4,DX
MOV AH,4CH
INT 21H
CODE ENDS
END MAIN
```

Program 2:

```
DATA SEGMENT
N1 DD 98EFEFEFH
N2 DD 11112222H
N3 DW ?
DATA ENDS
CODE SEGMENT
ASSUME CS:CODE, DS:DATA
START:
MOV AX,DATA
MOV DS,AX
MOV DL,00H
MOV AX, WORD PTR N1
MOV BX, WORD PTR N2
SUB AX,BX
MOV WORD PTR N3,AX
MOV AX, WORD PTR N1+2
MOV BX, WORD PTR N2+2
SBB AX,BX
MOV WORD PTR N3+2,AX
JNC MOVE
INC DL
MOVE: MOV BYTE PTR N3+4,DL
INT 3
CODE ENDS
END START
```

Output-Program1:

edit 32sub.asm

masm 32sub.asm

link 32sub.obj

debug 32sub.exe

-t (until 21H)

```
AX=4CCC  BX=2222  CX=003C  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=002A   NV UP EI PL ZR NA PE NC
076B:002A CD21          INT     21
-d 0000
076A:0000  BB BB EE EE 11 11 22 22-AA AA CC CC 00 00 00 00   ....""........
076A:0010  B8 6A 07 8E D8 BA 00 00-A1 00 00 8B 1E 04 00 F8   .j..............
076A:0020  2B C3 A3 08 00 A1 02 00-8B 1E 06 00 1B C3 A3 0A   +...............
076A:0030  00 83 DA 00 89 16 0C 00-B4 4C CD 21 56 FE 05 0C   .........L.!V...
076A:0040  00 52 50 E8 EA 48 83 C4-04 50 E8 7B 0E 83 C4 04   .RP..H...P.{....
```

Output-Program2:

edit 32sub1.asm

masm 32sub1.asm

link 32sub1.obj

debug 32sub1.exe

-r

-g

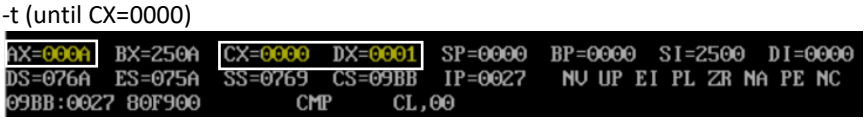
-r

```
AX=87DE  BX=1111  CX=0038  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=0027   NV UP EI NG NZ NA PE NC
076B:0027 CC            INT     3
-d 0000
076A:0000  EF EF EF 98 22 22 11 11-CD CD DE 87 00 00 00 00   ....""..........
076A:0010  B8 6A 07 8E D8 B2 00 A1-00 00 8B 1E 04 00 2B C3   .j..........+.
076A:0020  A3 08 00 A1 02 00 8B 1E-06 00 1B C3 A3 0A 00 73   ...............s
```

```
ASSUME DS:DATA,CS:CODE
DATA SEGMENT
ORG 2500H
ARR DW 0003,0005,0001,0010,0008
DATA ENDS
CODE SEGMENT
START:
MOV AX,DATA
MOV DS,AX
MOV SI, 2500H
MOV CX,0000
MOV CL,05
LEA BX,ARR
MOV DX,WORD PTR[BX]
MOV AX,0000
L1:CMP AX,WORD PTR[BX] ;largest
JNC L2
MOV AX,WORD PTR[BX]
L2:CMP DX,WORD PTR[BX] ;smallest
JC L3
MOV DX,WORD PTR[BX]
L3:ADD BX,02
DEC CL
CMP CL,00
JNZ L1
MOV AH,4CH
INT 21H
CODE ENDS
END START
```

Output:

```
edit lasm16.asm
masm lasm16.asm
link lasm16.obj
debug lasm16.exe
-t (until CX=0000)
```



```
DATA SEGMENT
MSG1 DB “HELLO$”
MSG2 DB “WORLD$”
DATA ENDS
ASSUME CS: CODE, DS: DATA
CODE SEGMENT
START:
MOV AX, DATA
MOV DS, AX
LEA DX, MSG1
MOV AH, 09H
INT 21H
LEA DX, MSG2
MOV AH, 09H
INT 21H
MOV AH, 4CH
INT 21H
CODE ENDS
END START
```

Output:

edit stringco.asm
masm stringco.asm
link stringco.obj
debug stringco.exe