

MASM Programs

BASIC ARITHMETIC OPERATIONS

STRING DISPLAY

STRING CONCATENATION

SORTING

SEARCHING

Commands to be followed in DOS box to run MASM programs

mount c c:\masm

edit pgmname.asm

masm pgmname.asm

link pgmname.obj

debug pgmname.exe

16 BIT ADDITION

ALGORITHM

Step I : Initialize the data segment.

Step II : Get the first number in AX register.

Step III : Get the second number in BX register.

Step IV : Add the two numbers.

Step V : Display the result.

PROGRAM

DATA SEGMENT

N1 DW 1731H

N2 DW 9212H

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

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
AX=076A BX=0000 CX=002C DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076B IP=0003  NU UP EI PL NZ NA PO NC
076B:0003 8ED8      MOV     DS,AX
-t
AX=076A BX=0000 CX=002C DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0005  NU UP EI PL NZ NA PO NC
076B:0005 33C0      XOR     AX,AX
-t
AX=0000 BX=0000 CX=002C DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0007  NU UP EI PL ZR NA PE NC
076B:0007 8BD8      MOV     BX,AX
-t
AX=0000 BX=0000 CX=002C DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0009  NU UP EI PL ZR NA PE NC
076B:0009 A10000     MOV     AX,[0000]          DS:0000=1731
-t
AX=1731 BX=0000 CX=002C DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=000C  NU UP EI PL ZR NA PE NC
076B:000C 03060200     ADD     AX,[0002]          DS:0002=9212
```

32 BIT ADDITION

ALGORITHM

Step II : Load the LSB of first number into AX register.

Step III : Load the MSB of first number into BX register.

Step IV : Load the LSB of the second number into CX register.

Step V : Load the MSB of the second number into DX register.

Step VI : Add the LSBs of two number.

-
- **PROGRAM**
- DATA SEGMENT
- LIST DD 12121212H,12121212H
- N3 DW ?
- N4 DW ?
- DATA ENDS
- CODE SEGMENT
- ASSUME CS :CODE;DS:DATA
- START:
- MOV AX,DATA
- MOV DS,AX
- XOR AX,AX
- MOV CL,AL
- MOV AX,[SI]
- ADD AX,[SI+4]
- MOV BX,AX
- MOV N3,BX
- MOV AX,[SI+2]

- ADD AX,[SI+6]
- MOV DX,AX
- MOV N4,DX
- JNC STOP
- INC CL
- STOP:
- MOV AX,4CH
- INT 21H
- CODE ENDS
- END START
- **OUTPUT**
-

```

AX=076A BX=0000 CX=0039 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076B IP=0003  NU UP EI PL NZ NA PO NC
076B:0003 8ED8          MOV     DS,AX
-t

AX=076A BX=0000 CX=0039 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0005  NU UP EI PL NZ NA PO NC
076B:0005 33C0          XOR     AX,AX
-t

AX=0000 BX=0000 CX=0039 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0007  NU UP EI PL ZR NA PE NC
076B:0007 8ACB          MOV     CL,AL
-t

AX=0000 BX=0000 CX=0000 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0009  NU UP EI PL ZR NA PE NC
076B:0009 8B04          MOV     AX,[SI]                      DS:0000=1212
-t

AX=1212 BX=0000 CX=0000 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=000B  NU UP EI PL ZR NA PE NC
076B:000B 034404       ADD     AX,[SI+04]          DS:0004=1212
-t

```

16 BIT SUBTRACTION

ALGORITHM

- Step I : Initialize the data segment.
 Step II : Get the first number in AX register.
 Step III : Get the second number in BX register.
 Step IV : Add the two numbers.
 Step V : Display the result.
 Step VI : Stop

PROGRAM

DATA SEGMENT

N1 DW 8888H

N2 DW 4444H

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
STOP:
MOV CX,AX
MOV AH,4CH
INT 21H
CODE ENDS
END START

```

OUTPUT

```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
AX=076A BX=0000 CX=002C DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076B IP=0003  NU UP EI PL NZ NA PO NC
076B:0003 8ED8      MOV     DS,AX
-
-t
AX=076A BX=0000 CX=002C DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0005  NU UP EI PL NZ NA PO NC
076B:0005 33C0      XOR     AX,AX
-t
AX=0000 BX=0000 CX=002C DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0007  NU UP EI PL ZR NA PE NC
076B:0007 8BD8      MOV     BX,AX
-t
AX=0000 BX=0000 CX=002C DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0009  NU UP EI PL ZR NA PE NC
076B:0009 A10000     MOV     AX,[0000]      DS:0000=8888
-t
AX=8888 BX=0000 CX=002C DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=000C  NU UP EI PL ZR NA PE NC
076B:000C 2B060200     SUB     AX,[0002]      DS:0002=4444
-

```

32 BIT SUBTRACTION

ALGORITHM

- Step I : Initialize the data segment.
- Step II : Load the LSB of first number into AX register.
- Step III : Load the MSB of first number into BX register.
- Step IV : Load the LSB of the second number into CX register.
- Step V : Load the MSB of the second number into DX register.

PROGRAM

```

DATA SEGMENT
LIST DD 12121212H,12121212H
N3 DW ?
N4 DW ?
DATA ENDS
CODE SEGMENT
ASSUME CS :CODE;DS:DATA

```

START:

```
MOV AX,DATA
MOV DS,AX
XOR AX,AX
MOV CL,AL
MOV AX,[SI]
ADD AX,[SI+4]
MOV BX,AX
MOV N3,BX
MOV AX,[SI+2]
ADD AX,[SI+6]
MOV DX,AX
MOV N4,DX
JNC STOP
```

INC CL

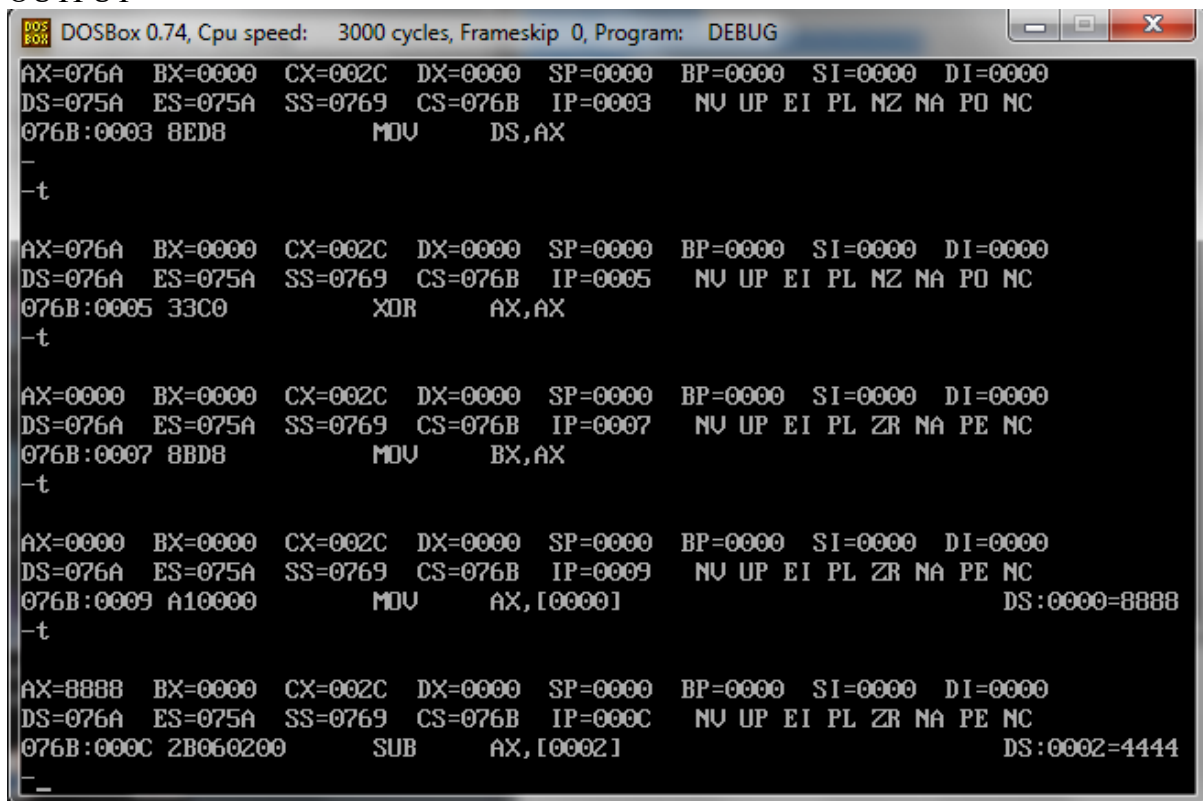
STOP:

```
MOV AX,4CH
INT 21H
```

CODE ENDS

END START

OUTPUT



The screenshot shows the DOSBox 0.74 DEBUG window with the following state and execution steps:

Step	Address	Instruction	AX	BX	CX	DX	SP	BP	SI	DI	DS	ES	SS	CS	IP	Flags	Comments
1	076B:0003	MOV DS,AX	076A	0000	002C	0000	0000	0000	0000	0000	075A	075A	0769	076B	0003	NU UP EI PL NZ NA PO NC	
2	076B:0005	XOR AX,AX	076A	0000	002C	0000	0000	0000	0000	0000	076A	075A	0769	076B	0005	NU UP EI PL NZ NA PO NC	
3	076B:0007	MOV BX,AX	0000	0000	002C	0000	0000	0000	0000	0000	076A	075A	0769	076B	0007	NU UP EI PL ZR NA PE NC	
4	076B:0009	MOV AX,[0000]	0000	0000	002C	0000	0000	0000	0000	0000	076A	075A	0769	076B	0009	NU UP EI PL ZR NA PE NC	DS:0000=8888
5	076B:000C	SUB AX,[0002]	8888	0000	002C	0000	0000	0000	0000	0000	076A	075A	0769	076B	000C	NU UP EI PL ZR NA PE NC	DS:0002=4444

16 BIT MULTIPLICATION

ALGORITHM

- Step I : Initialize the data segment.
- Step II : Get the first number in AX register.
- Step III : Get the second number in BX register.
- Step IV : Multiply the two 16 bit numbers.
- Step V : Display the result.
- Step VI : Stop

PROGRAM

DATA SEGMENT

N1 DW 8888H

N2 DW 4444H

N3 DW ?

DATA ENDS

CODE SEGMENT

ASSUME CS :CODE;DS:DATA

START:

MOV AX,4343

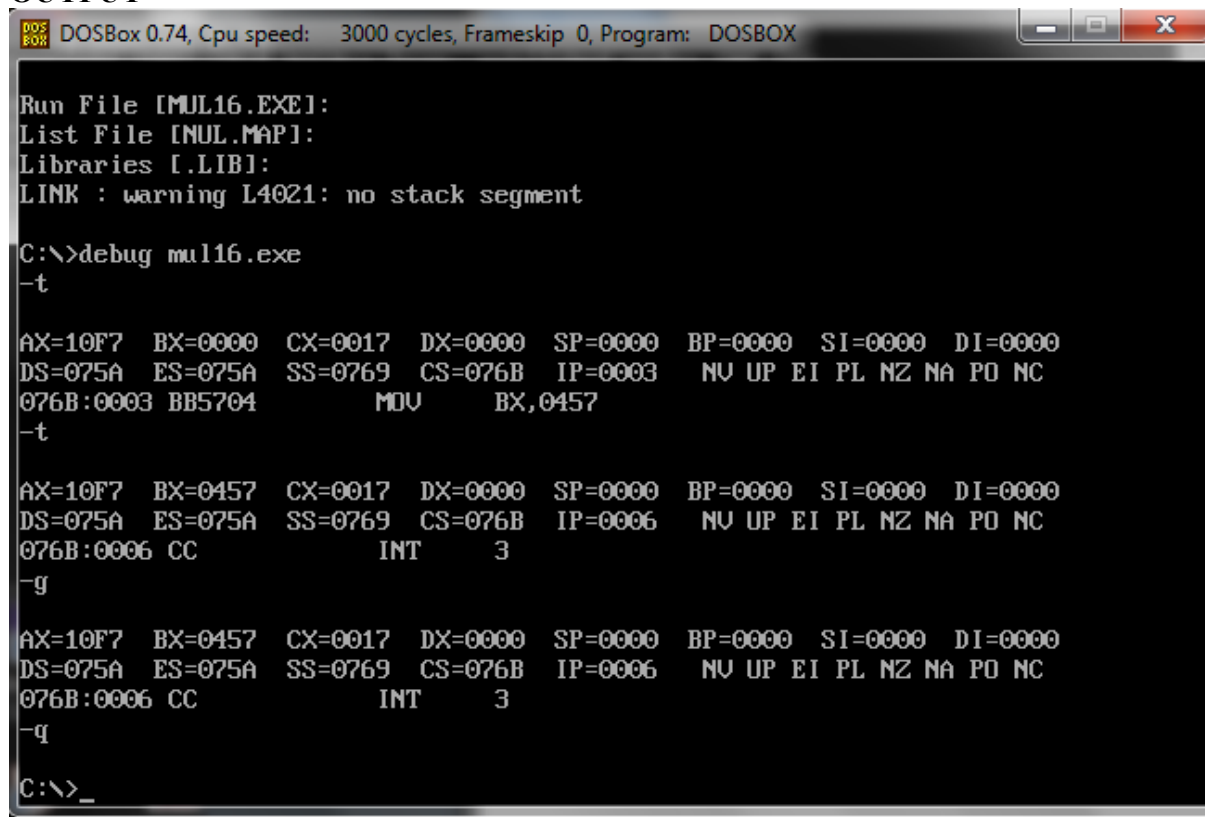
MOV BX,1111

INT 3

CODE ENDS

END START

OUTPUT



```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

Run File [MUL16.EXE]:
List File [MUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\>debug mul16.exe
-t

AX=10F7 BX=0000 CX=0017 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076B IP=0003  NV UP EI PL NZ NA PO NC
076B:0003 BB5704      MOV     BX,0457
-t

AX=10F7 BX=0457 CX=0017 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076B IP=0006  NV UP EI PL NZ NA PO NC
076B:0006 CC          INT     3
-g

AX=10F7 BX=0457 CX=0017 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076B IP=0006  NV UP EI PL NZ NA PO NC
076B:0006 CC          INT     3
-q

C:\>_
```

STRING DISPLAY

AIM

Write a program to display a given string

PROGRAM

DATA SEGMENT

MSG1 DB "HELLO WORLD\$"

DATA ENDS

ASSUME CS:CODE; DS:DATA

CODE SEGMENT

START:

MOV AX,DATA

MOV DS,AX

```

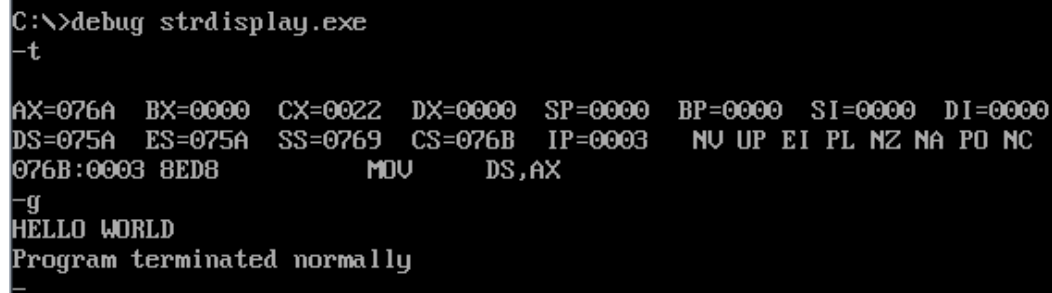
MOV DX,OFFSET MSG1
MOV AH,09H
INT 21H
MOV AH,4CH
MOV AL,00H
INT 21H

```

CODE ENDS

END START

OUTPUT



```

C:\>debug strdisplay.exe
-t
AX=076A  BX=0000  CX=0022  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076B  IP=0003  NU UP EI PL NZ NA PO NC
076B:0003 8ED8          MOV     DS,AX
-g
HELLO WORLD
Program terminated normally
-

```

STRING CONCATENATION

AIM

Write a program to concatenate two strings

PROGRAM

DATA SEGMENT

MSG1 DB "HELLO\$"

MSG2 DB "WORLD\$"

DATA ENDS

ASSUME CS:CODE; DS:DATA

CODE SEGMENT

START:

MOV AX,DATA

MOV DS,AX

MOV DX,OFFSET MSG1

MOV AH,09H

INT 21H

MOV DX,OFFSET MSG2

MOV AH,09H

INT 21H

CODE ENDS

END START

OUTPUT

```
DOS BOX DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.
Run File [STRCONCAT.EXE]:
List File [INUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\>debug strconcat.exe
-t
AX=076A BX=0000 CX=0023 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076B IP=0003  NU UP EI PL NZ NA PO NC
076B:0003 8ED8      MOV     DS,AX
-t
AX=076A BX=0000 CX=0023 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0005  NU UP EI PL NZ NA PO NC
076B:0005 BA0000    MOV     DX,0000
-t
AX=076A BX=0000 CX=0023 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0008  NU UP EI PL NZ NA PO NC
076B:0008 B409      MOV     AH,09
-g
HELLOWORLD_
```

SORTING

AIM

Write a program to perform sorting

PROGRAM

DATA SEGMENT

STRING1 DB 99H,12H,56H,45H,36H

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

START: MOV AX,DATA

MOV DS,AX

MOV CH,04H

UP2: MOV CL,04H

LEA SI,STRING1

UP1:MOV AL,[SI]

MOV BL,[SI+1]

CMP AL,BL

JNC DOWN

MOV DL,[SI+1]

XCHG [SI],DL

MOV [SI+1],DL

DOWN: INC SI

DEC CL

JNZ UP1

DEC CH

JNZ UP2

INT 3

CODE ENDS

END START

OUTPUT

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
-t
AX=076A BX=0000 CX=003B DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0005  NU UP EI PL NZ NA PO NC
076B:0005 B504      MOV     CH,04
-t
AX=076A BX=0000 CX=043B DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0007  NU UP EI PL NZ NA PO NC
076B:0007 B104      MOV     CL,04
-g
AX=072D BX=000C CX=0000 DX=002D SP=0000 BP=0000 SI=0004 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0027  NU UP EI PL ZR NA PE NC
076B:0027 CC        INT     3
-d
076B:0000 B8 6A 07 8E D8 B5 04 B1-04 8D 36 00 00 8A 04 8A  .j.....6....
076B:0010 5C 01 3A C3 73 08 8A 54-01 86 14 88 54 01 46 FE  \.:.s..T...T.F.
076B:0020 C9 75 EA FE CD 75 E0 CC-8B 46 FC 8B 56 FE 05 0C  .u...u...F..U...
076B:0030 00 52 50 E8 EA 48 83 C4-04 50 E8 7B 0E 83 C4 04  .RP..H...P.f....
076B:0040 3D FF FF 74 03 E9 ED 00-C4 5E FC 26 8A 47 0C 2A  =..t.....^..&.G.*
076B:0050 E4 40 50 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83  .@P.....RP..H.
076B:0060 C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6  ..P....P..s.....
076B:0070 FA FE 81 E6 FF 00 C6 82-FB FE 00 2B C0 50 8D 86  ....+..P..
```

SEARCHING

AIM

Write a program to perform searching

PROGRAM

DATA SEGMENT

STRING1 DB 11H,22H,33H,44H,55H

MSG1 DB "FOUND\$"

MSG2 DB "NOT FOUND\$"

SE DB 33H

DATA ENDS

PRINT MACRO MSG

MOV AH, 09H

LEA DX, MSG

INT 21H

INT 3

ENDM

CODE SEGMENT

ASSUME CS:CODE, DS:DATA

START:

MOV AX, DATA

MOV DS, AX

MOV AL, SE

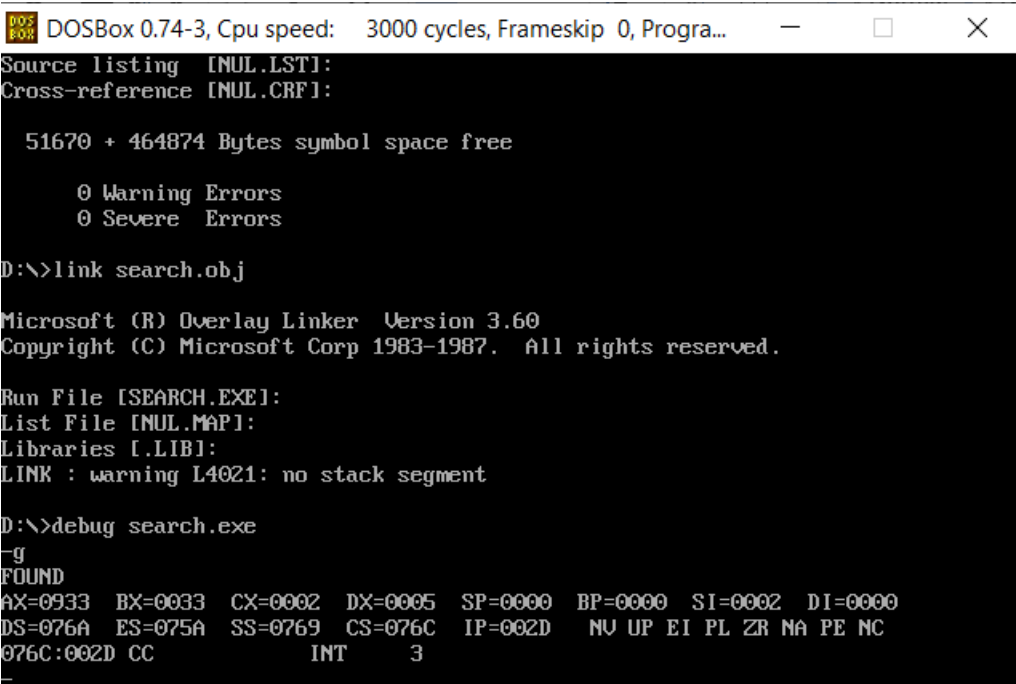
```
LEA SI, STRING1
MOV CX, 04H
```

```
UP:
MOV BL,[SI]
CMP AL, BL
JZ FO
INC SI
DEC CX
JNZ UP
PRINT MSG2
JMP END1
```

```
FO:
PRINT MSG1
```

```
END1:
INT 3
CODE ENDS
END START
```

OUTPUT

A screenshot of a DOSBox window titled "DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...". The window has a black background with white text. The text shows the output of a linker and a debugger. It includes source listing and cross-reference information for INUL.LST1 and INUL.CRF1, indicating 51670 + 464874 Bytes symbol space free and 0 Warning Errors and 0 Severe Errors. It then shows the command "D:\>link search.obj" followed by the Microsoft (R) Overlay Linker Version 3.60 copyright notice. Next, it shows "Run File [SEARCH.EXE]:", "List File [INUL.MAP]:", "Libraries [I.LIB]:", and a "LINK : warning L4021: no stack segment". Finally, it shows the command "D:\>debug search.exe" followed by a list of registers and their values: AX=0933, BX=0033, CX=0002, DX=0005, SP=0000, BP=0000, SI=0002, DI=0000, DS=076A, ES=075A, SS=0769, CS=076C, IP=002D, and a status line showing "076C:002D CC INT 3".

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
Source listing [INUL.LST1:
Cross-reference [INUL.CRF1:

51670 + 464874 Bytes symbol space free

0 Warning Errors
0 Severe Errors

D:\>link search.obj

Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

Run File [SEARCH.EXE]:
List File [INUL.MAP]:
Libraries [I.LIB]:
LINK : warning L4021: no stack segment

D:\>debug search.exe
-g
FOUND
AX=0933 BX=0033 CX=0002 DX=0005 SP=0000 BP=0000 SI=0002 DI=0000
DS=076A ES=075A SS=0769 CS=076C IP=002D NU UP EI PL ZR NA PE NC
076C:002D CC INT 3
```