VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Fall Semester 2021-22

Microprocessors and Interfacing LAB
CSE2006
Slot – L43+L44
Digital Assignment 3

Name: Ishan Sagar Jogalekar

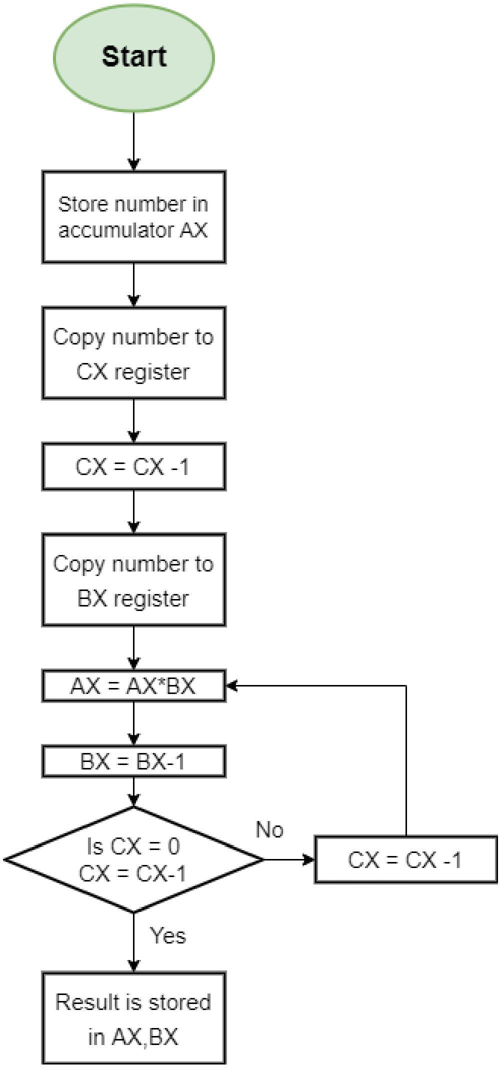
Reg. No: 19BCE2250

1. Factorial of number -

Input:

Here value 9 as input. The value 9 is being stored in location 0000H. We are using n as variable data byte with value = 9 so that factorial of 9 will be calculated.

Flowchart:

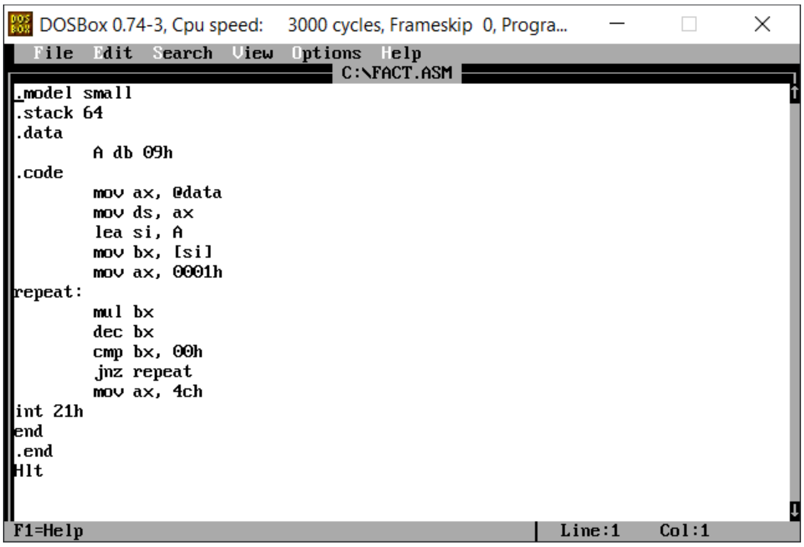

Start
Store number in accumulator AX
Copy number to CX register
CX = CX -1
Copy number to BX register
AX = AX*BX
BX = BX-1
Is CX = 0
CX = CX-1
No
CX = CX -1
Yes
Result is stored in AX,BX

Algorithm:

1) Input the Number whose factorial is to be find and Store that Number in CX Register (Condition for LOOP Instruction).
2) Insert 0001 in AX (Condition for MUL Instruction) and 0000 in DX.
3) Multiply CX with AX until CX become Zero (0) using LOOP Instruction.
4) Copy the content of AX to memory location 0600.
5) Copy the content of DX to memory location 0601.
6) Stop Execution.

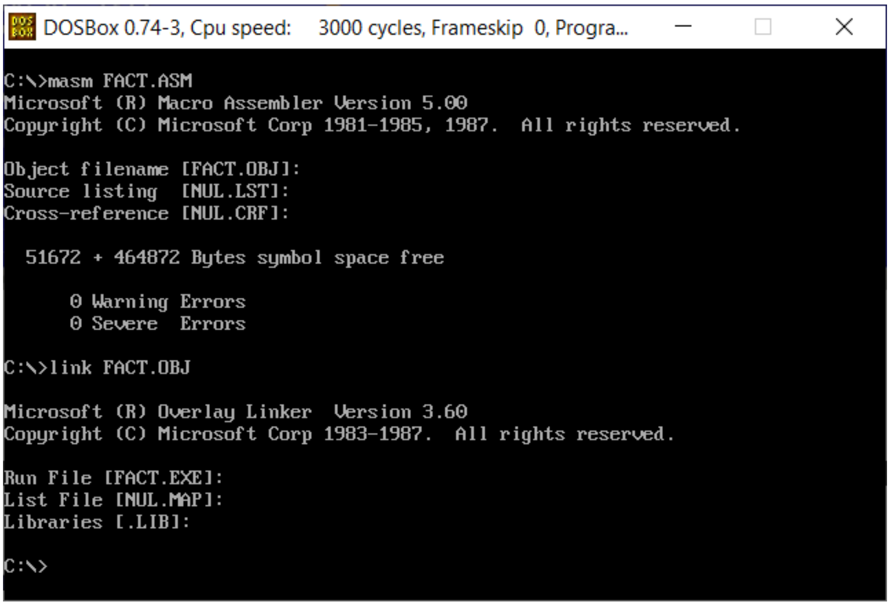
Program:

```
.model small
.stack 64
.data
        A db 09h
.code
        mov ax, @data
        mov ds, ax
        lea si, A
        mov bx, [si]
        mov ax, 0001h
repeat:
        mul bx
        dec bx
        cmp bx, 00h
        jnz repeat
        mov ax, 4ch
int 21h
end
.end
Hlt
```

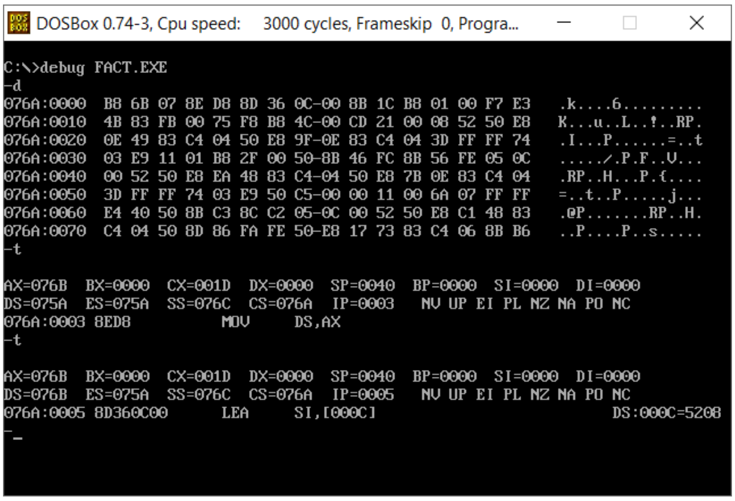

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
File Edit Search View Options Help
C:\FACT.ASM
F1=Help
Line:1 Col:1

```
.model small
.stack 64
.data
        A db 09h
.code
        mov ax, @data
        mov ds, ax
        lea si, A
        mov bx, [si]
        mov ax, 0001h
repeat:
        mul bx
        dec bx
        cmp bx, 00h
        jnz repeat
        mov ax, 4ch
int 21h
end
.end
Hlt
```

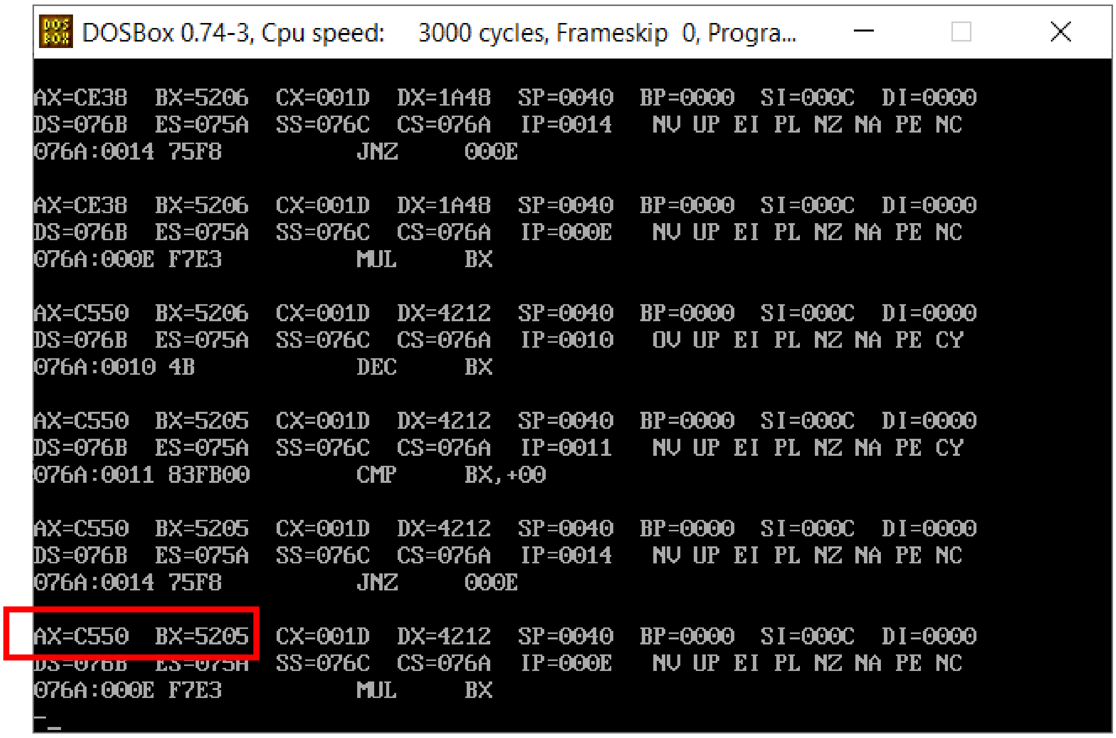
Output:


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

```
C:\>masm FACT.ASM
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [FACT.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51672 + 464872 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\>link FACT.OBJ

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

Run File [FACT.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:

C:\>
```


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

```
C:\>debug FACT.EXE
-d
076A:0000  B8 6B 07 8E D8 8D 36 0C-00 8B 1C B8 01 00 F7 E3   .k....6.........
076A:0010  4B 83 FB 00 75 F8 B8 4C-00 CD 21 00 08 52 50 E8   K...u..L..!..RP.
076A:0020  0E 49 83 C4 04 50 E8 9F-0E 83 C4 04 3D FF FF 74   .I...P......=..t
076A:0030  03 E9 11 01 B8 2F 00 50-8B 46 FC 8B 56 FE 05 0C   ...../.P.F..V...
076A:0040  00 52 50 E8 EA 48 83 C4-04 50 E8 7B 0E 83 C4 04   .RP..H...P.{....
076A:0050  3D FF FF 74 03 E9 50 C5-00 00 11 00 6A 07 FF FF   =..t..P.....j...
076A:0060  E4 40 50 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83   .@P.......RP..H.
076A:0070  C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6   ..P....P..s.....
-t

AX=076B  BX=0000  CX=001D  DX=0000  SP=0040  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=076C  CS=076A  IP=0003   NV UP EI PL NZ NA PO NC
076A:0003 8ED8          MOV     DS,AX
-t

AX=076B  BX=0000  CX=001D  DX=0000  SP=0040  BP=0000  SI=0000  DI=0000
DS=076B  ES=075A  SS=076C  CS=076A  IP=0005   NV UP EI PL NZ NA PO NC
076A:0005 8D360C00      LEA     SI,[000C]                          DS:000C=5208
-_
```


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

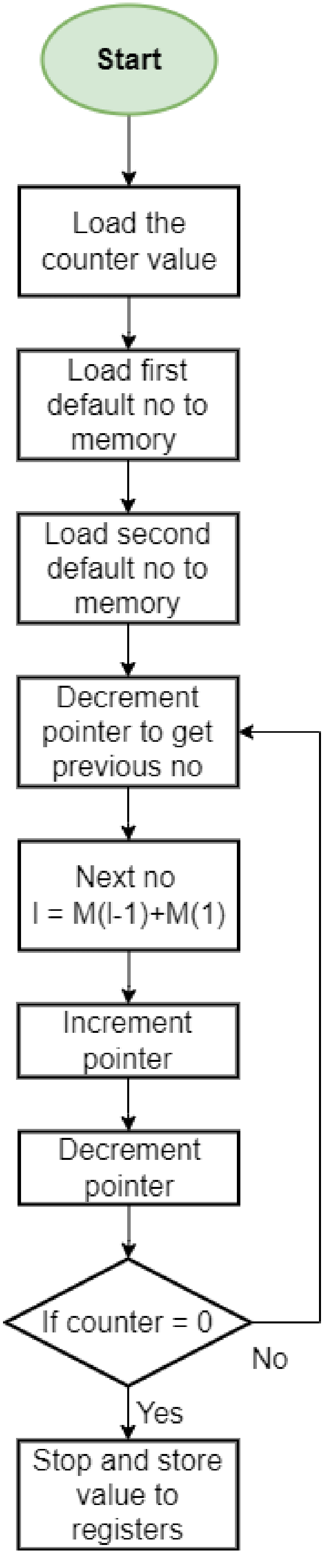
```
AX=CE38  BX=5206  CX=001D  DX=1A48  SP=0040  BP=0000  SI=000C  DI=0000
DS=076B  ES=075A  SS=076C  CS=076A  IP=0014   NV UP EI PL NZ NA PE NC
076A:0014 75F8          JNZ     000E

AX=CE38  BX=5206  CX=001D  DX=1A48  SP=0040  BP=0000  SI=000C  DI=0000
DS=076B  ES=075A  SS=076C  CS=076A  IP=000E   NV UP EI PL NZ NA PE NC
076A:000E F7E3          MUL     BX

AX=C550  BX=5206  CX=001D  DX=4212  SP=0040  BP=0000  SI=000C  DI=0000
DS=076B  ES=075A  SS=076C  CS=076A  IP=0010   OV UP EI PL NZ NA PE CY
076A:0010 4B            DEC     BX

AX=C550  BX=5205  CX=001D  DX=4212  SP=0040  BP=0000  SI=000C  DI=0000
DS=076B  ES=075A  SS=076C  CS=076A  IP=0011   NV UP EI PL NZ NA PE CY
076A:0011 83FB00        CMP     BX,+00

AX=C550  BX=5205  CX=001D  DX=4212  SP=0040  BP=0000  SI=000C  DI=0000
DS=076B  ES=075A  SS=076C  CS=076A  IP=0014   NV UP EI PL NZ NA PE NC
076A:0014 75F8          JNZ     000E

AX=C550  BX=5205  CX=001D  DX=4212  SP=0040  BP=0000  SI=000C  DI=0000
DS=076B  ES=075A  SS=076C  CS=076A  IP=000E   NV UP EI PL NZ NA PE NC
076A:000E F7E3          MUL     BX
-_
```

2. Fibonacci series –

Input:

Initialize counter with number of terms in series and load it to DB. Also considering default numbers as input 0H and 01H as starting of any Fibonacci series.

Flowchart:

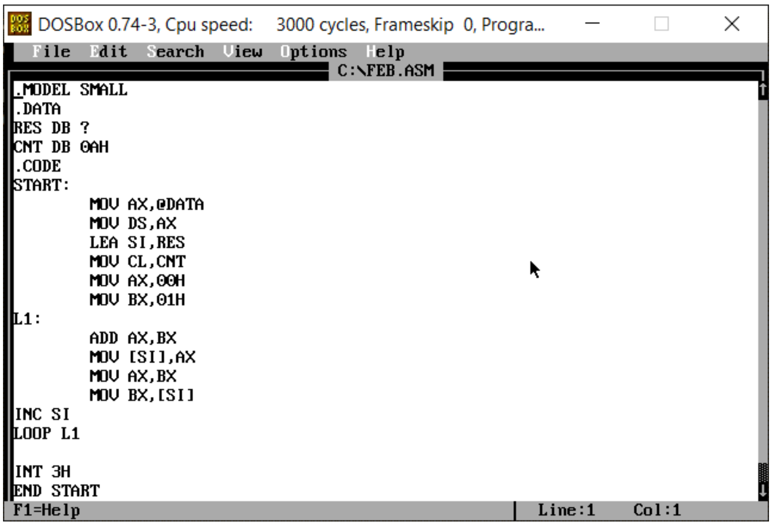

Start
Load the counter value
Load first default no to memory
Load second default no to memory
Decrement pointer to get previous no
Next no
I = M(I-1)+M(1)
Increment pointer
Decrement pointer
If counter = 0
No
Yes
Stop and store value to registers

Algorithm:

1. Move the value stored at offset 00H into CX- consider to be counter, and decrement it by 2.
2. Move 00H into AL
3. Move 500 into SI
4. Move AL into [SI]
5. Increment both AL and SI by 1, and store AL's value in [SI].
6. Move [SI-1]th value into AL
7. Move [SI]th value into AH
8. Move 00H into BH
9. Add BH and AH
10. Add BH again with AL
11. Increment SI by 1
12. Store BH into [SI]
13. Loop back to Step 6 till counter becomes 0
14. Stop

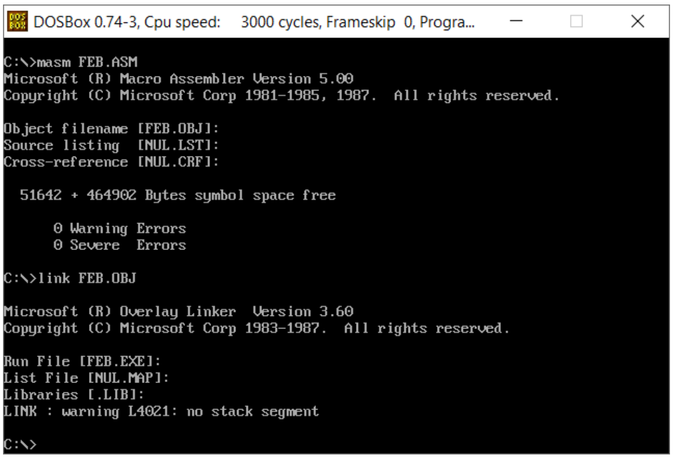
Program:

```
.MODEL SMALL

.DATA

RES DB ?

CNT DB 0AH

.CODE

START:

        MOV AX,@DATA

        MOV DS,AX

        LEA SI,RES

        MOV CL,CNT

        MOV AX,00H

        MOV BX,01H

L1:

        ADD AX,BX

        MOV [SI],AX

        MOV AX,BX

        MOV BX,[SI]
        INC SI
LOOP L1
INT 3H
END START
```

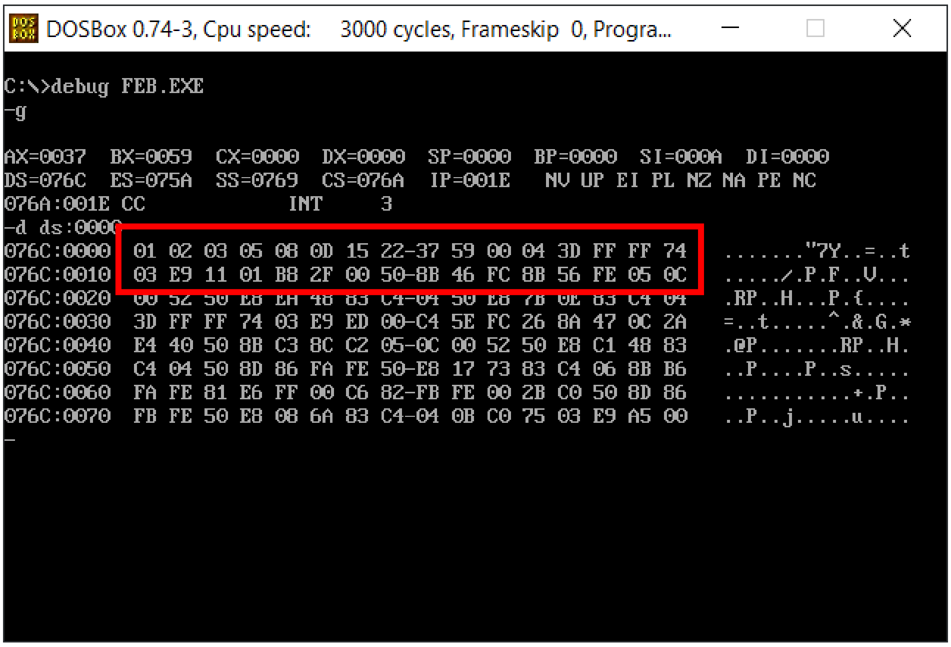

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
File Edit Search View Options Help
C:\FEB.ASM
F1=Help
Line:1   Col:1

```
.MODEL SMALL
.DATA
RES DB ?
CNT DB 0AH
.CODE
START:
        MOV AX,@DATA
        MOV DS,AX
        LEA SI,RES
        MOV CL,CNT
        MOV AX,00H
        MOV BX,01H
L1:
        ADD AX,BX
        MOV [SI],AX
        MOV AX,BX
        MOV BX,[SI]
INC SI
LOOP L1

INT 3H
END START
```

Output:


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

```
C:\>masm FEB.ASM
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [FEB.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51642 + 464902 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\>link FEB.OBJ

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

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

C:\>
```


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

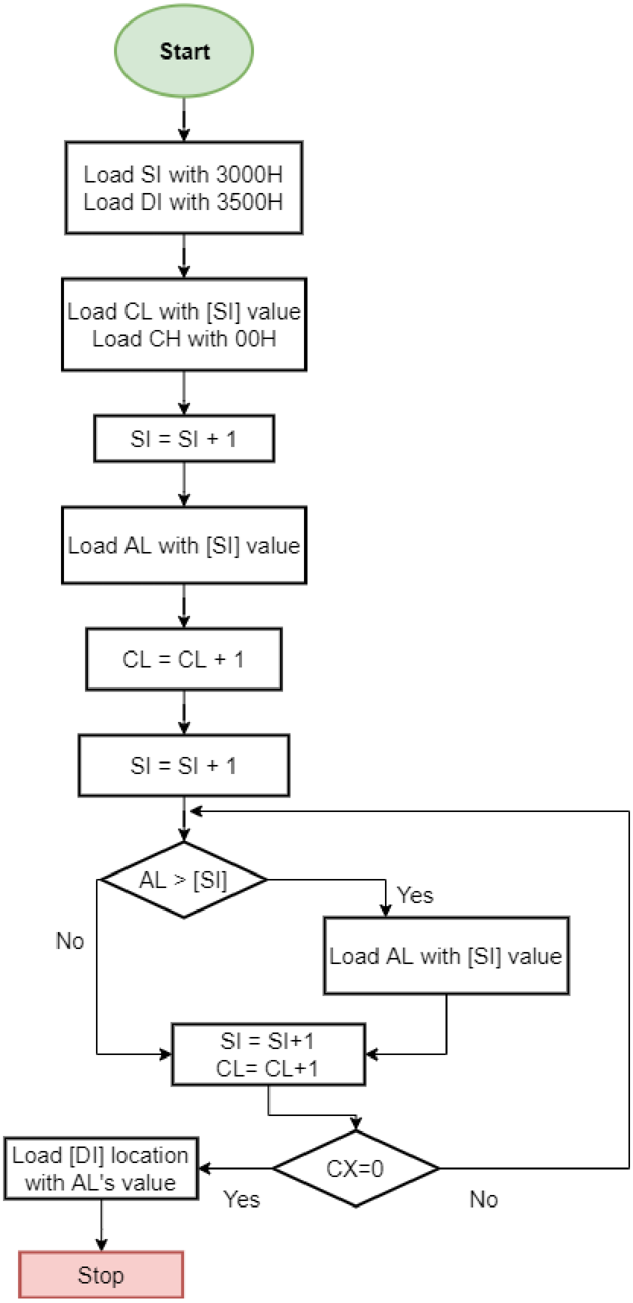
```
C:\>debug FEB.EXE
-g

AX=0037  BX=0059  CX=0000  DX=0000  SP=0000  BP=0000  SI=000A  DI=0000
DS=076C  ES=075A  SS=0769  CS=076A  IP=001E   NV UP EI PL NZ NA PE NC
076A:001E CC            INT     3
-d ds:0000
076C:0000  01 02 03 05 08 0D 15 22-37 59 00 04 3D FF FF 74   ......."7Y..=..t
076C:0010  03 E9 11 01 B8 2F 00 50-8B 46 FC 8B 56 FE 05 0C   ...../.P.F..V...
076C:0020  00 52 50 E8 EA 48 83 C4-04 50 E8 7B 0E 83 C4 04   .RP..H...P.{....
076C:0030  3D FF FF 74 03 E9 ED 00-C4 5E FC 26 8A 47 0C 2A   =..t.....^.&.G.*
076C:0040  E4 40 50 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83   .@P.......RP..H.
076C:0050  C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6   ..P....P..s.....
076C:0060  FA FE 81 E6 FF 00 C6 82-FB FE 00 2B C0 50 8D 86   ...........+.P..
076C:0070  FB FE 50 E8 08 6A 83 C4-04 0B C0 75 03 E9 A5 00   ..P..j.....u....
-
```

3.1 Largest number in given numbers-

Input:

Considering input of numbers in string initially stored in DB and dynamically stored in subsequent registers.

Flowchart:



Algorithm:

1. Initialize base address of the data segment. Using the displacements from this base address, store values in different registers.
2. Store offset 3000 in SI-register&offset3500inDIregister.

3. Load data from SI to register CL and set register CH to 00.
4. Load first number from next offset to register AL and decrease CL by 1.
5. Compare value of register AL with the data at next offset.
6. If that data is greater than value of register AL then update value of register AL to that data else no changes are required.
7. Increase the offset value for next comparison and decrease count by 1 and continue this till CL becomes 0. Store result, largest number stored in AL register, to offset address 3500, which is pointed to by DI register. Mov result to register BX.

Program:

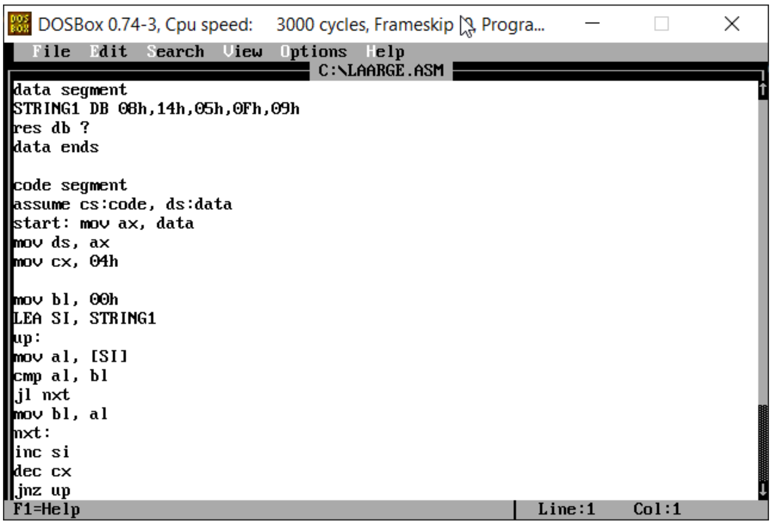
```
data segment

STRING1 DB 08h,14h,05h,0Fh,09h

res db ?

data ends


code segment

assume cs:code, ds:data

start: mov ax, data

mov ds, ax

mov cx, 04h


mov bl, 00h

LEA SI, STRING1

up:

mov al, [SI]

cmp al, bl

jl nxt

mov bl, al

nxt:

inc si

dec cx

jnz up
```

mov res,bl

int 3

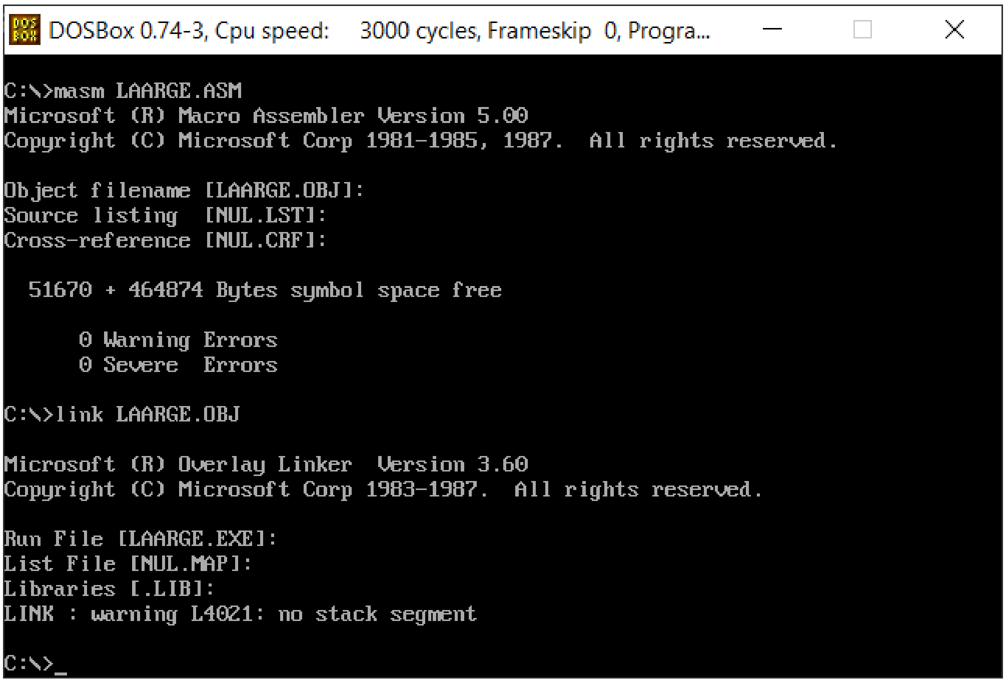
code ends

end start

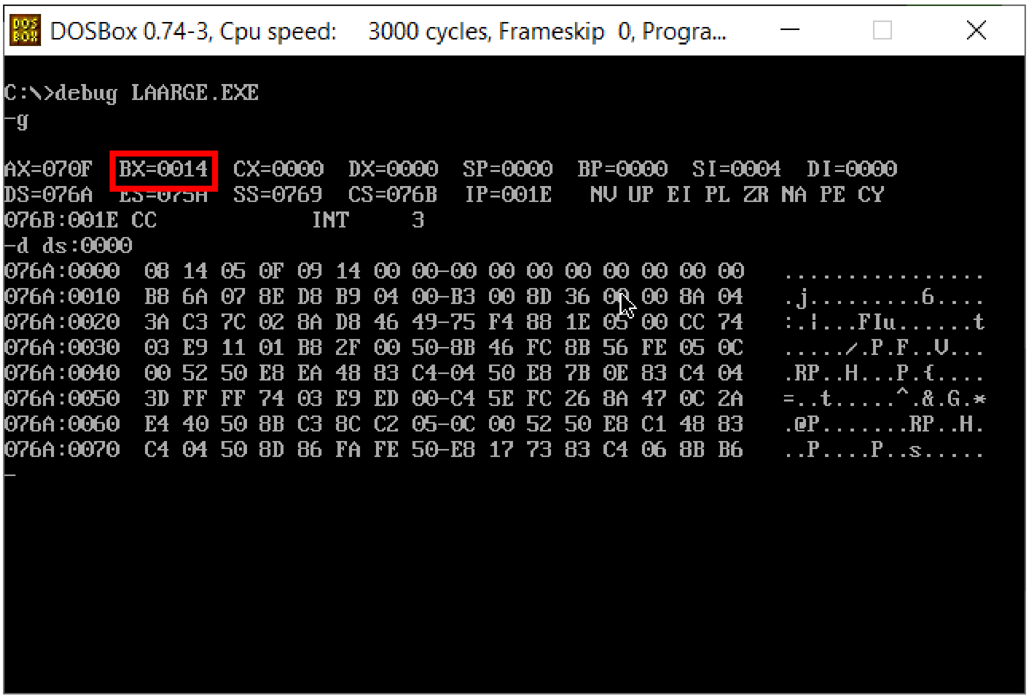

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
File Edit Search View Options Help
C:\LAARGE.ASM
F1=Help
Line:1 Col:1

```
data segment
STRING1 DB 08h,14h,05h,0Fh,09h
res db ?
data ends

code segment
assume cs:code, ds:data
start: mov ax, data
mov ds, ax
mov cx, 04h

mov bl, 00h
LEA SI, STRING1
up:
mov al, [SI]
cmp al, bl
jl nxt
mov bl, al
nxt:
inc si
dec cx
jnz up
```

Output:


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

```
C:\>masm LAARGE.ASM
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [LAARGE.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51670 + 464874 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\>link LAARGE.OBJ

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

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

C:\>_
```


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

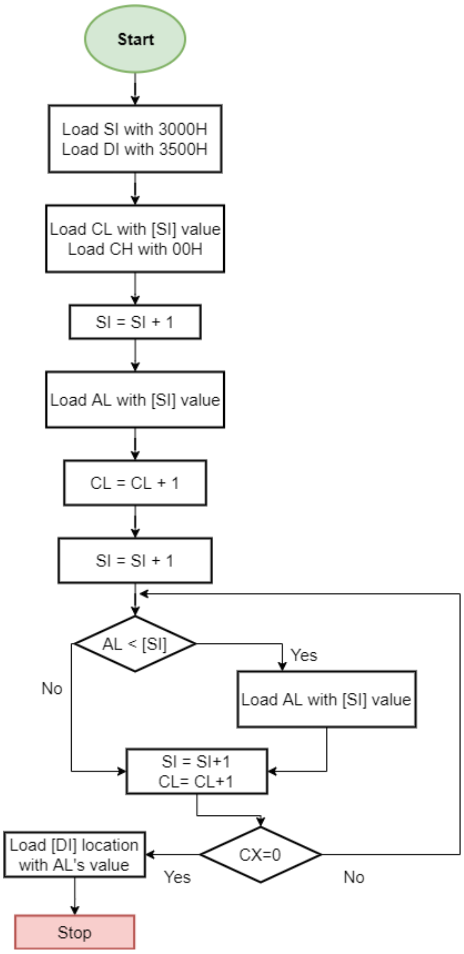
```
C:\>debug LAARGE.EXE
-g

AX=070F  BX=0014  CX=0000  DX=0000  SP=0000  BP=0000  SI=0004  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=001E   NV UP EI PL ZR NA PE CY
076B:001E CC            INT     3
-d ds:0000
076A:0000  08 14 05 0F 09 14 00 00-00 00 00 00 00 00 00 00   ................
076A:0010  B8 6A 07 8E D8 B9 04 00-B3 00 8D 36 00 00 8A 04   .j.........6....
076A:0020  3A C3 7C 02 8A D8 46 49-75 F4 88 1E 05 00 CC 74   :.|...FIu......t
076A:0030  03 E9 11 01 B8 2F 00 50-8B 46 FC 8B 56 FE 05 0C   ...../.P.F..V...
076A:0040  00 52 50 E8 EA 48 83 C4-04 50 E8 7B 0E 83 C4 04   .RP..H...P.{....
076A:0050  3D FF FF 74 03 E9 ED 00-C4 5E FC 26 8A 47 0C 2A   =..t.....^.&.G.*
076A:0060  E4 40 50 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83   .@P.......RP..H.
076A:0070  C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6   ..P....P..s.....
-
```

3.2 Smallest number in given numbers-

Input:

Considering input of numbers in string initially stored in DB and dynamically stored in subsequent registers.

Flowchart:



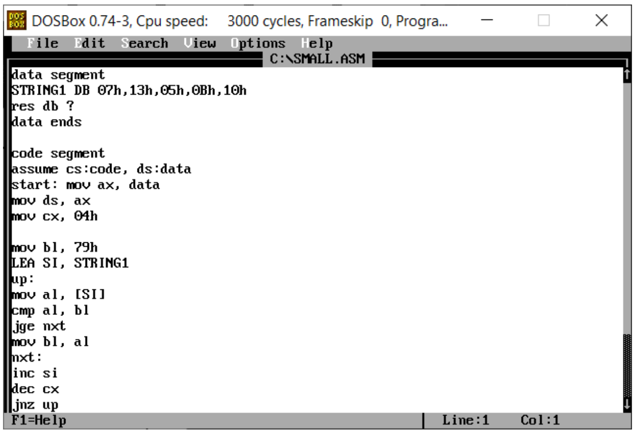
Algorithm:

1. Initialize base address of the data segment. Using the displacements from this base address, store values in different registers.
2. Store offset 3000 in SI-register&offset3500inDIregister.
3. Load data from SI to register CL and set register CH to 00.
4. Load first number from next offset to register AL and decrease CL by 1.
5. Compare value of register AL with the data at next offset.
6. If that data is smaller than value of register AL then update value of register AL to that data else no changes are required.
7. Increase the offset value for next comparison and decrease count by 1 and continue this till CL becomes 0. Store result, largest number stored in AL register, to offset address 3500, which is pointed to by DI register. Mov result to BX register.

Program:

```
data segment

STRING1 DB 07h,13h,05h,0Bh,10h

res db ?

data ends

code segment

assume cs:code, ds:data

start: mov ax, data

mov ds, ax

mov cx, 04h

mov bl, 79h

LEA SI, STRING1

up:

mov al, [SI]

cmp al, bl

jge nxt

mov bl, al

nxt:

inc si

dec cx
```

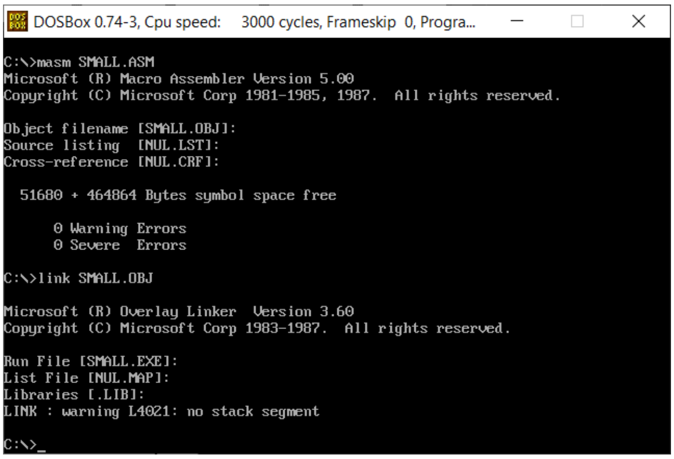
```
        jnz up

        mov res,bl

        int 3

        code ends

end start
```



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
File Edit Search View Options Help
C:\SMALL.ASM

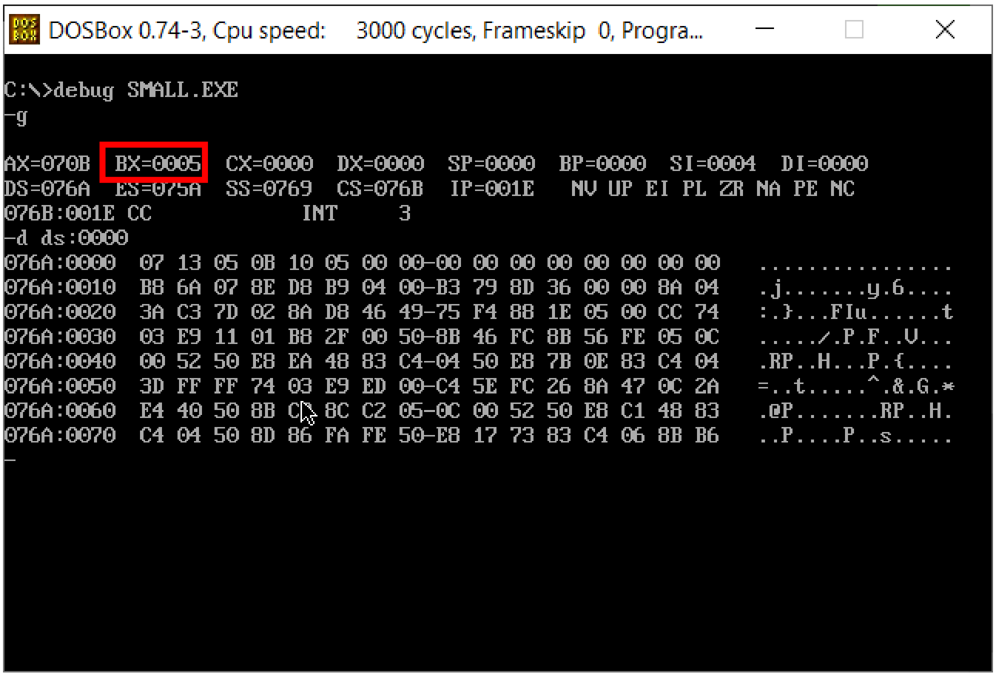
```
data segment
STRING1 DB 07h,13h,05h,0Bh,10h
res db ?
data ends

code segment
assume cs:code, ds:data
start: mov ax, data
mov ds, ax
mov cx, 04h

mov bl, 79h
LEA SI, STRING1
up:
mov al, [SI]
cmp al, bl
jge nxt
mov bl, al
nxt:
inc si
dec cx
jnz up
```

Output:



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

```
C:\>masm SMALL.ASM
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [SMALL.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51680 + 464864 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\>link SMALL.OBJ

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

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

C:\>_
```



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

```
C:\>debug SMALL.EXE
-g

AX=070B  BX=0005  CX=0000  DX=0000  SP=0000  BP=0000  SI=0004  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=001E   NV UP EI PL ZR NA PE NC
076B:001E CC            INT     3
-d ds:0000
076A:0000  07 13 05 0B 10 05 00 00-00 00 00 00 00 00 00 00   ................
076A:0010  B8 6A 07 8E D8 B9 04 00-B3 79 8D 36 00 00 8A 04   .j.......y.6....
076A:0020  3A C3 7D 02 8A D8 46 49-75 F4 88 1E 05 00 CC 74   :.}...FIu......t
076A:0030  03 E9 11 01 B8 2F 00 50-8B 46 FC 8B 56 FE 05 0C   ...../.P.F..V...
076A:0040  00 52 50 E8 EA 48 83 C4-04 50 E8 7B 0E 83 C4 04   .RP..H...P.{....
076A:0050  3D FF FF 74 03 E9 ED 00-C4 5E FC 26 8A 47 0C 2A   =..t.....^.&.G.*
076A:0060  E4 40 50 8B [illegible] 8C C2 05-0C 00 52 50 E8 C1 48 83   .@P.......RP..H.
076A:0070  C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6   ..P....P..s.....
-
```

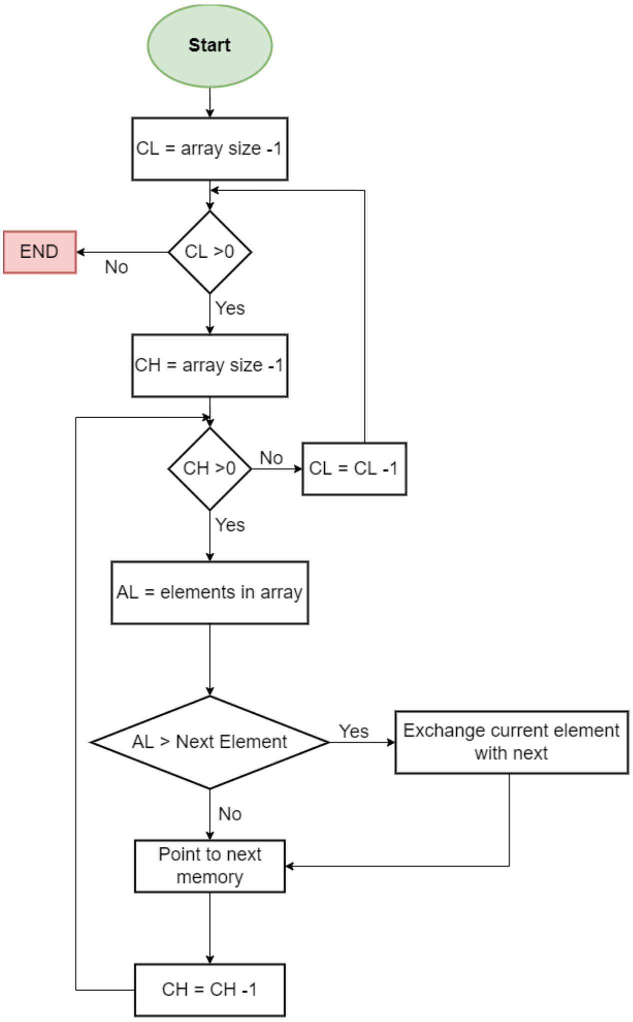
4.Bubble sort –

A. Ascending Order:

Input:

supplying 16-bit input in the data segment by creating an array called list using assemble directive method. Here we are using Data Word (DW) which means that we are using core digit value.

Flowchart:



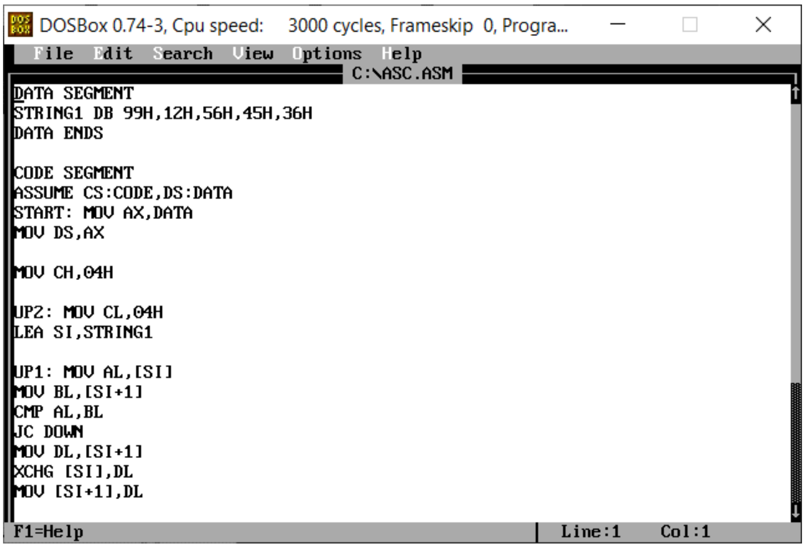
Algorithm:

1. Load data from offset 500 to register CL (for count).
2. Travel from starting memory location to last and compare two numbers
3. If first number is greater than second number then swap them.
4. First pass fixes the position for last number.
5. Decrease the count by 1.
6. Again, travel from starting memory location to (last-1, by help of count) and compare two numbers
7. If first number is greater than second number then swap them.
8. Second pass fix the position for last two numbers.

9. Repeat.

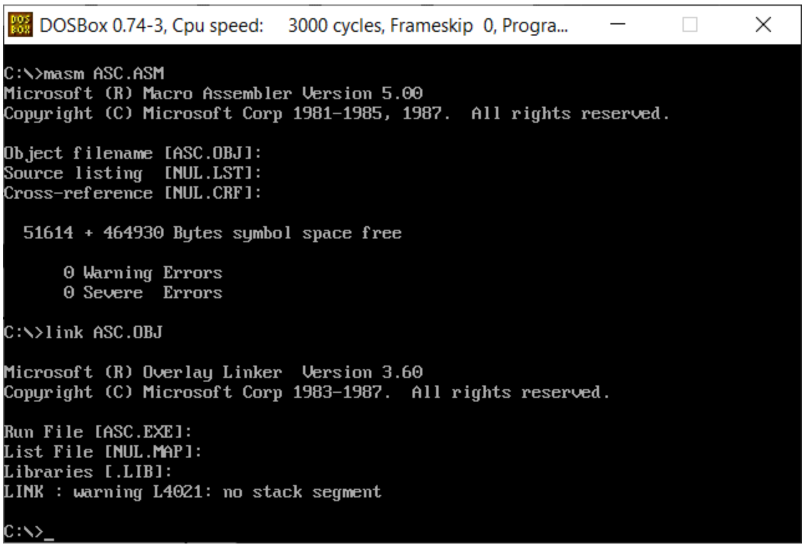
Program:

```
DATA SEGMENT
STRING1 DB 19H,10H,50H,04H,32H
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
JC 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
```

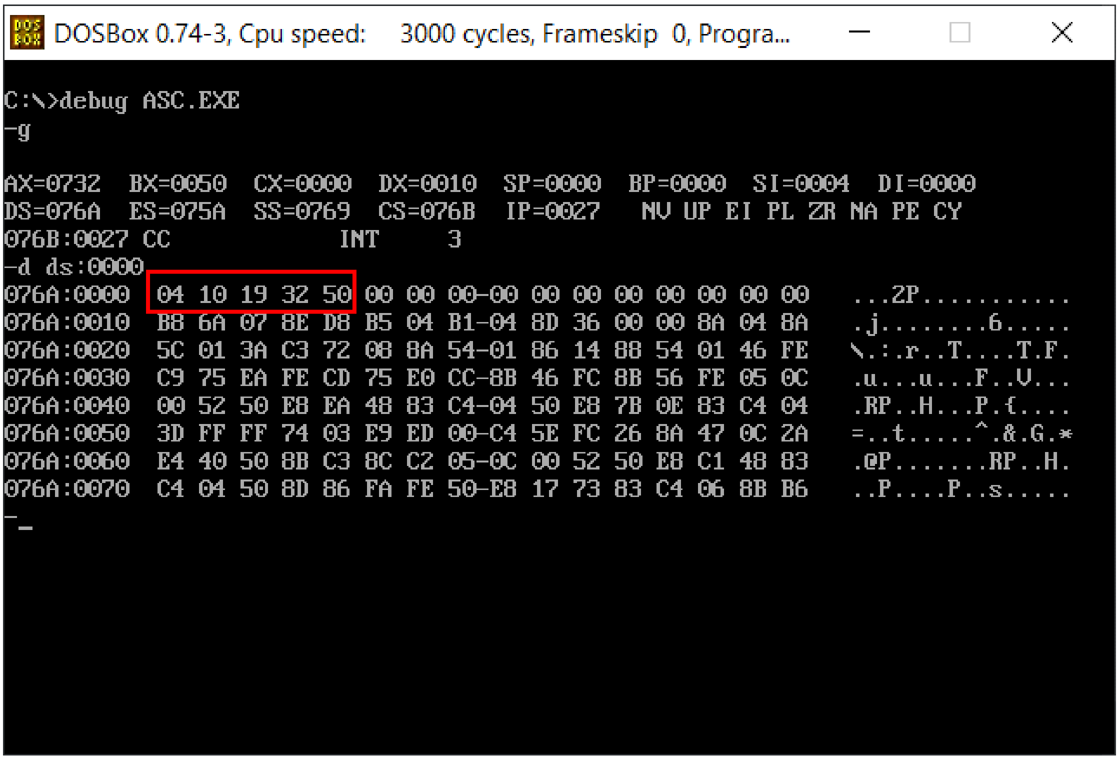

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
File Edit Search View Options Help
C:\ASC.ASM
F1=Help
Line:1 Col:1

```
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
JC DOWN
MOV DL,[SI+1]
XCHG [SI],DL
MOV [SI+1],DL
```

Output:


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

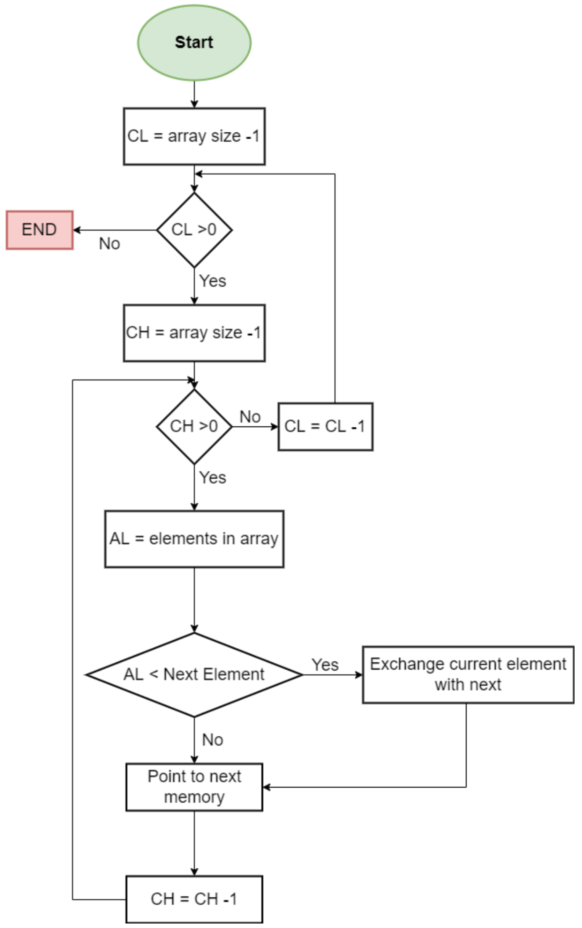
```
C:\>masm ASC.ASM
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [ASC.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51614 + 464930 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\>link ASC.OBJ

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

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

C:\>_
```


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

```
C:\>debug ASC.EXE
-g

AX=0732  BX=0050  CX=0000  DX=0010  SP=0000  BP=0000  SI=0004  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=0027   NV UP EI PL ZR NA PE CY
076B:0027 CC            INT     3
-d ds:0000
076A:0000  04 10 19 32 50 00 00 00-00 00 00 00 00 00 00 00   ...2P...........
076A:0010  B8 6A 07 8E D8 B5 04 B1-04 8D 36 00 00 8A 04 8A   .j........6.....
076A:0020  5C 01 3A C3 72 08 8A 54-01 86 14 88 54 01 46 FE   \.:.r..T....T.F.
076A:0030  C9 75 EA FE CD 75 E0 CC-8B 46 FC 8B 56 FE 05 0C   .u...u...F..V...
076A:0040  00 52 50 E8 EA 48 83 C4-04 50 E8 7B 0E 83 C4 04   .RP..H...P.{....
076A:0050  3D FF FF 74 03 E9 ED 00-C4 5E FC 26 8A 47 0C 2A   =..t.....^.&.G.*
076A:0060  E4 40 50 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83   .@P.......RP..H.
076A:0070  C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6   ..P....P..s.....
-_
```

B. Descending Order:

Flowchart:



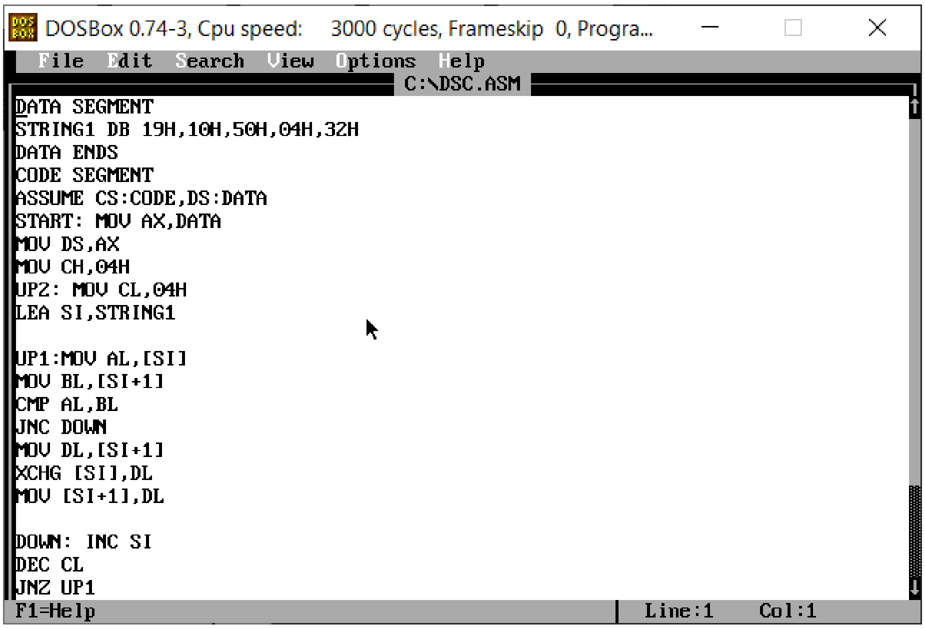
Algorithm:

1. Load data from offset 500 to register CL (for count).
2. Travel from starting memory location to last and compare two numbers
3. If first number is less than second number then swap them.
4. First pass fixes the position for last number.
5. Decrease the count by 1.
6. Again, travel from starting memory location to (last-1, by help of count) and compare two numbers
7. If first number is less than second number then swap them.
8. Second pass fix the position for last two numbers.
9. Repeat.

Program:

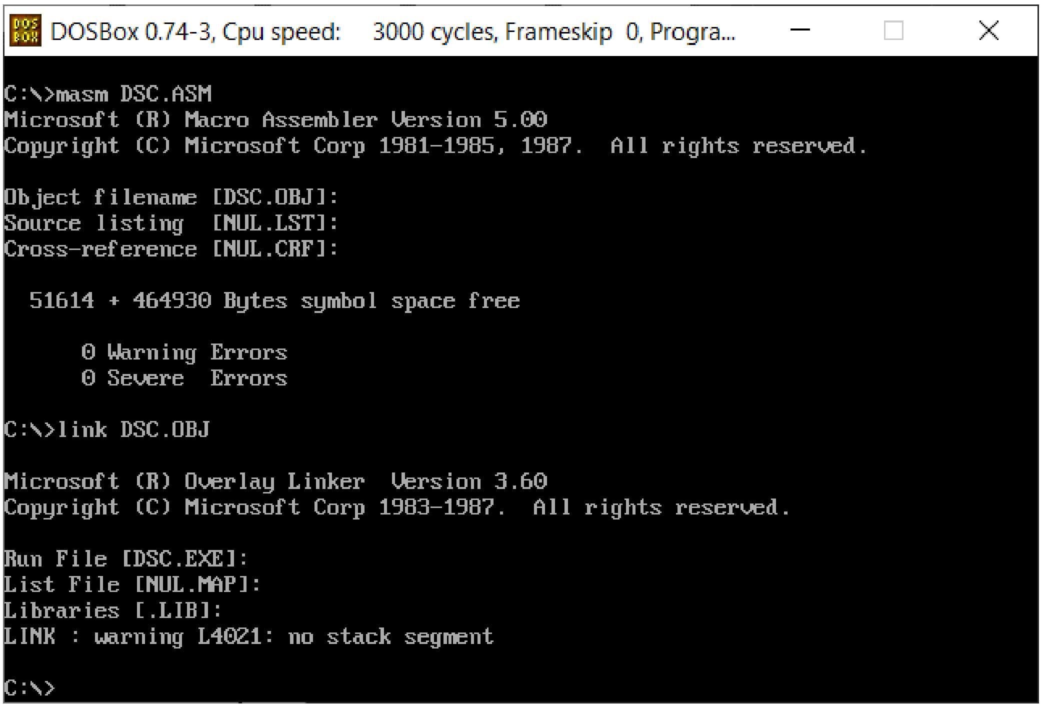
```
DATA SEGMENT

STRING1 DB 19H,10H,50H,04H,32H

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
```


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
File Edit Search View Options Help
C:\DSC.ASM
F1=Help
Line:1 Col:1

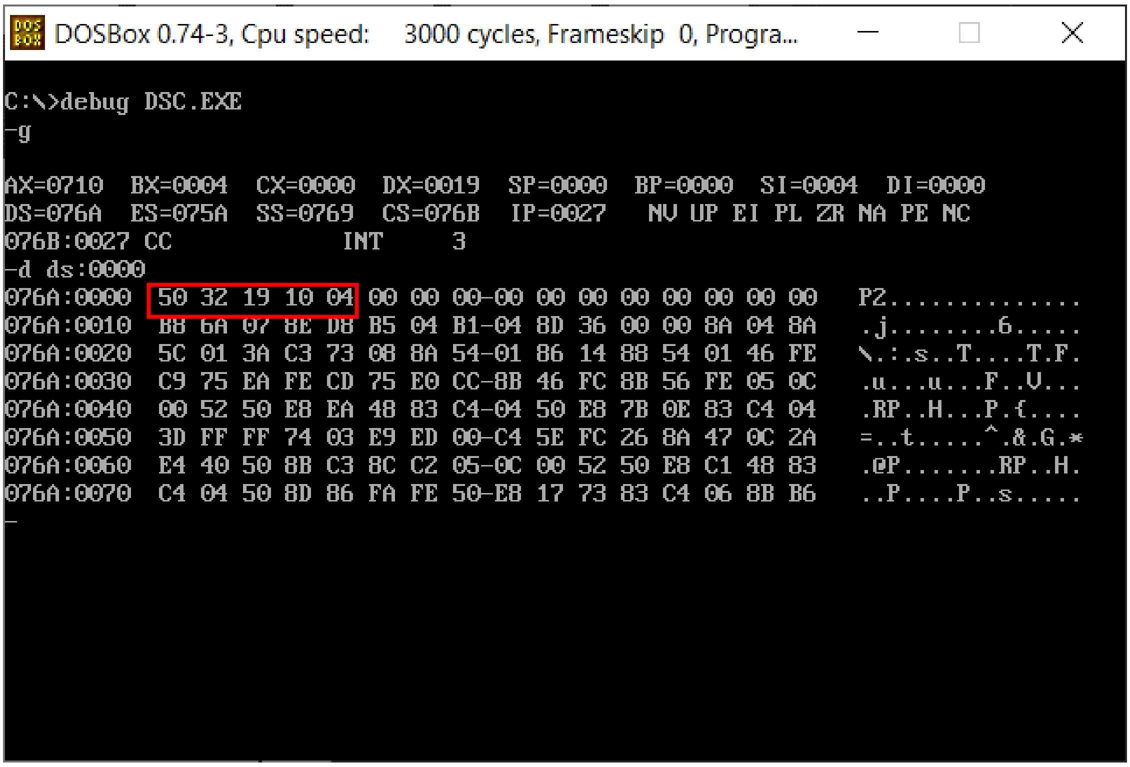
```
DATA SEGMENT
STRING1 DB 19H,10H,50H,04H,32H
DATA ENDS
CODE SEGMENT
ASSUME CS:CODE,DS:DATA
START: MOV AX,DATA
MOV DS,AX
MOV CH,04H
UP2: MOV CL,04H
LEA SI,STRING1

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
```

Output:



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

```
C:\>masm DSC.ASM
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [DSC.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51614 + 464930 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\>link DSC.OBJ

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

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

C:\>
```



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

```
C:\>debug DSC.EXE
-g

AX=0710  BX=0004  CX=0000  DX=0019  SP=0000  BP=0000  SI=0004  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=0027   NV UP EI PL ZR NA PE NC
076B:0027 CC            INT     3
-d ds:0000
076A:0000  50 32 19 10 04 00 00 00-00 00 00 00 00 00 00 00   P2..............
076A:0010  B8 6A 07 8E D8 B5 04 B1-04 8D 36 00 00 8A 04 8A   .j........6.....
076A:0020  5C 01 3A C3 73 08 8A 54-01 86 14 88 54 01 46 FE   \.:.s..T....T.F.
076A:0030  C9 75 EA FE CD 75 E0 CC-8B 46 FC 8B 56 FE 05 0C   .u...u...F..V...
076A:0040  00 52 50 E8 EA 48 83 C4-04 50 E8 7B 0E 83 C4 04   .RP..H...P.{....
076A:0050  3D FF FF 74 03 E9 ED 00-C4 5E FC 26 8A 47 0C 2A   =..t.....^.&.G.*
076A:0060  E4 40 50 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83   .@P.......RP..H.
076A:0070  C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6   ..P....P..s.....
-
```