



Microprocessor, Embedded Systems and IoT

LAB REPORT

Course Code: CSE 232

Submitted to:

Sonia Nasrin

Lecturer

Daffodil International University

Date of Submission:
12/8/2022

Submitted by:

MD. RAKIBUL ISLAM SHANTO

ID no: 203-15-3871

Section: PC-A

Department of CSE

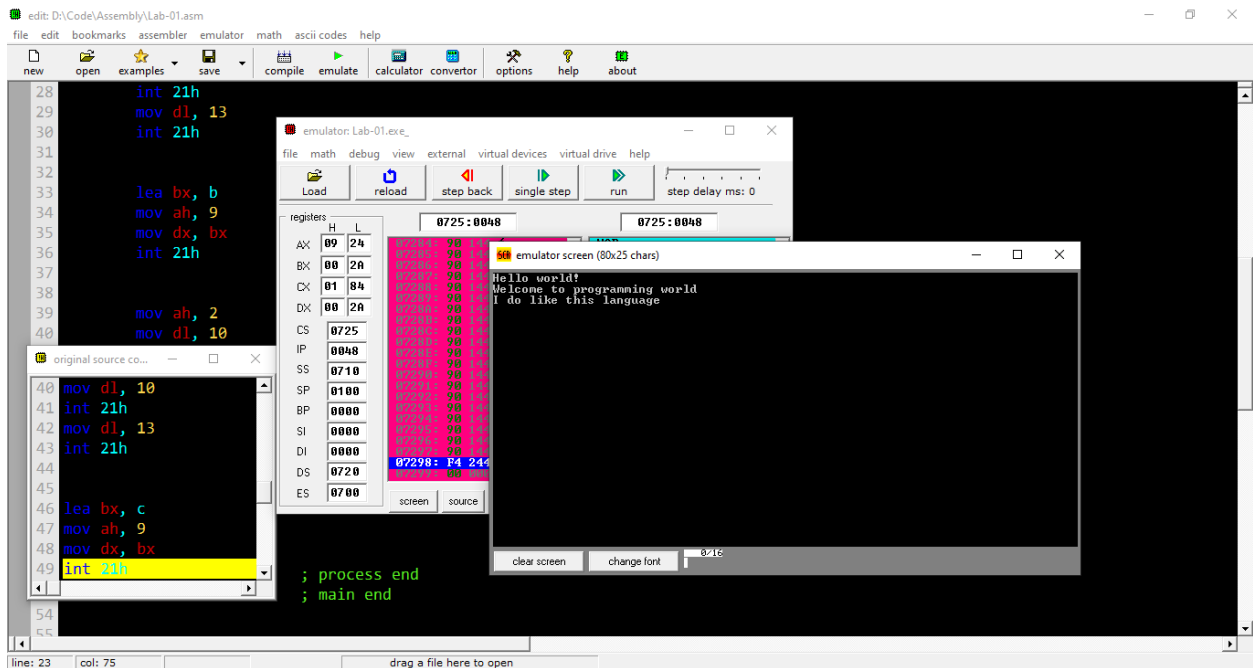
Daffodil International University



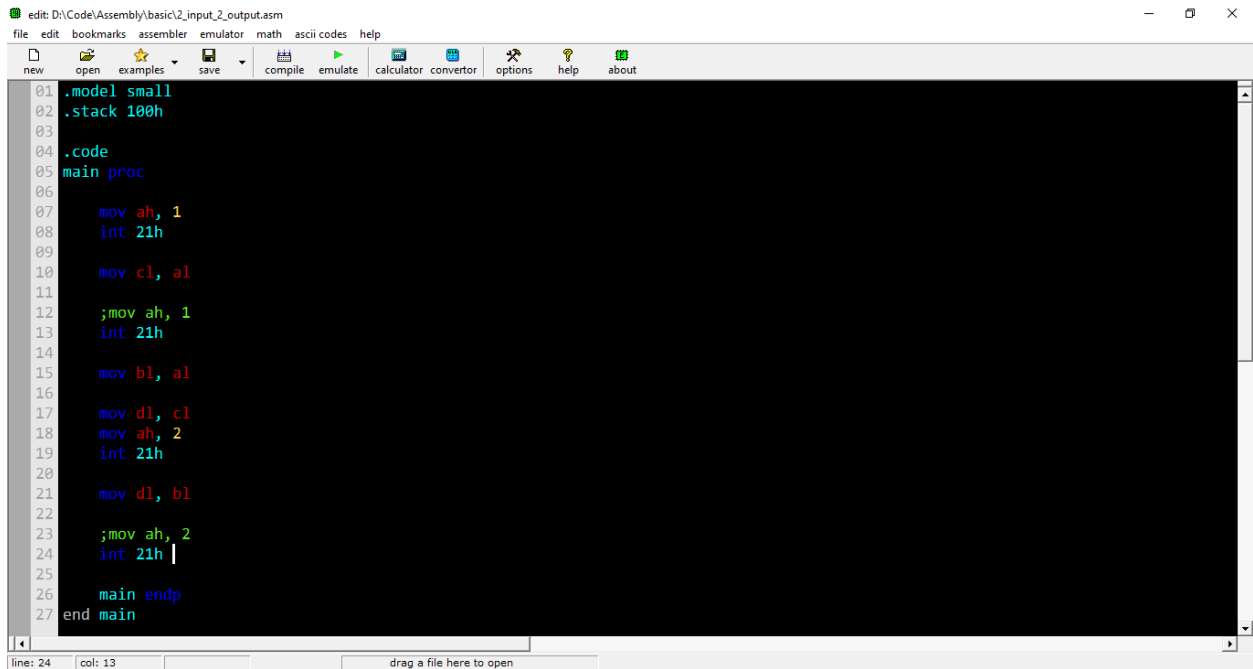
Printing 3 strings from data segment

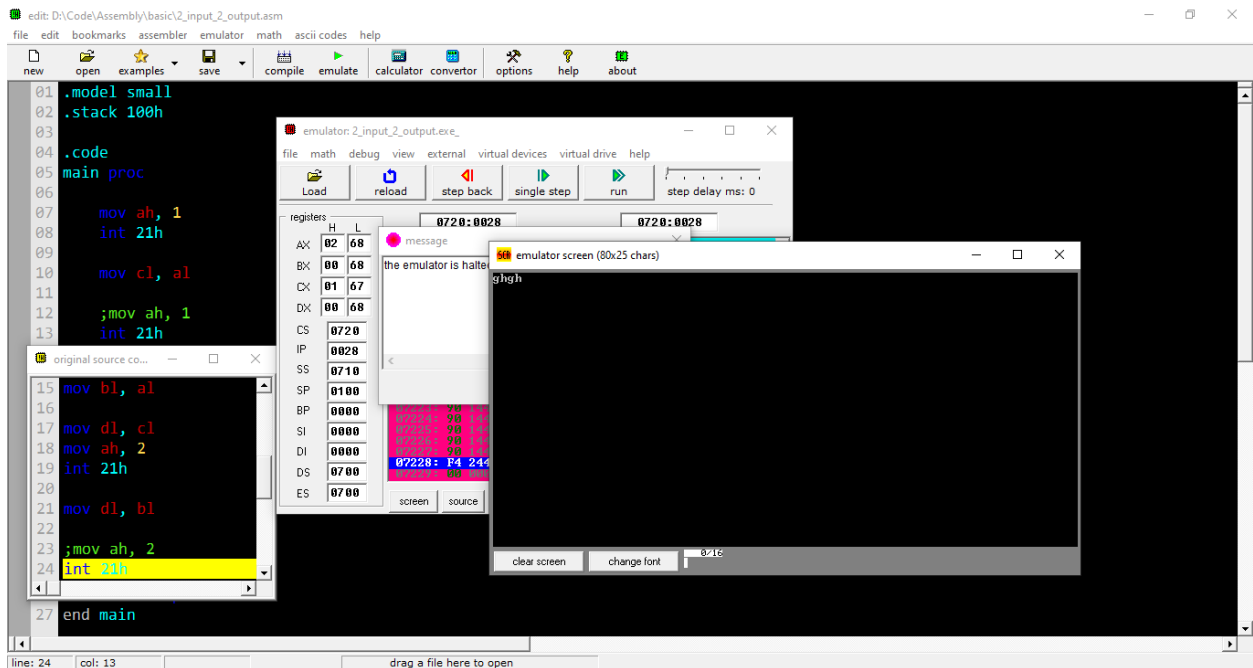
```
edit: D:\Code\Assembly\Lab-01.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
01 ;include 'emu8086.inc' ; include studio.h -> calling header file which can take standard input + output
02
03 .model small ; code / data segment
04 .stack 100h ; store + size h,d,b
05 .data ; Data segment (ds) variable declaration
06
07 a db "Hello world!$"
08 b db "Welcome to programming world$"
09 c db "I do like this language$"
10
11 .code ; Code segment (cs)
12 main proc ; main () <- C ; proc -> process
13
14 ; print "Hello world!" ; when u call a headerfile
15
16
17 mov ax, @data ; ax -> accumulator reg. (input/output) ax,bx,cx -> memory
18 mov ds, ax ; to get access from ds to cs
19
20 lea bx, a ; lea -> to load a string a -> bx
21 mov ah, 9 ; to print the string
22 mov dx, bx ; dx -> print
23 int 21h ; interrupt.....print will work now
24
25
26 mov ah, 2
27 mov dl, 10 ; new lines
28 int 21h
29
```

```
edit: D:\Code\Assembly\Lab-01.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
28 int 21h
29 mov dl, 13
30 int 21h
31
32
33 lea bx, b
34 mov ah, 9
35 mov dx, bx
36 int 21h
37
38
39 mov ah, 2
40 mov dl, 10
41 int 21h
42 mov dl, 13
43 int 21h
44
45
46 lea bx, c
47 mov ah, 9
48 mov dx, bx
49 int 21h
50
51
52 main endp ; process end
53 end main ; main end
54
```

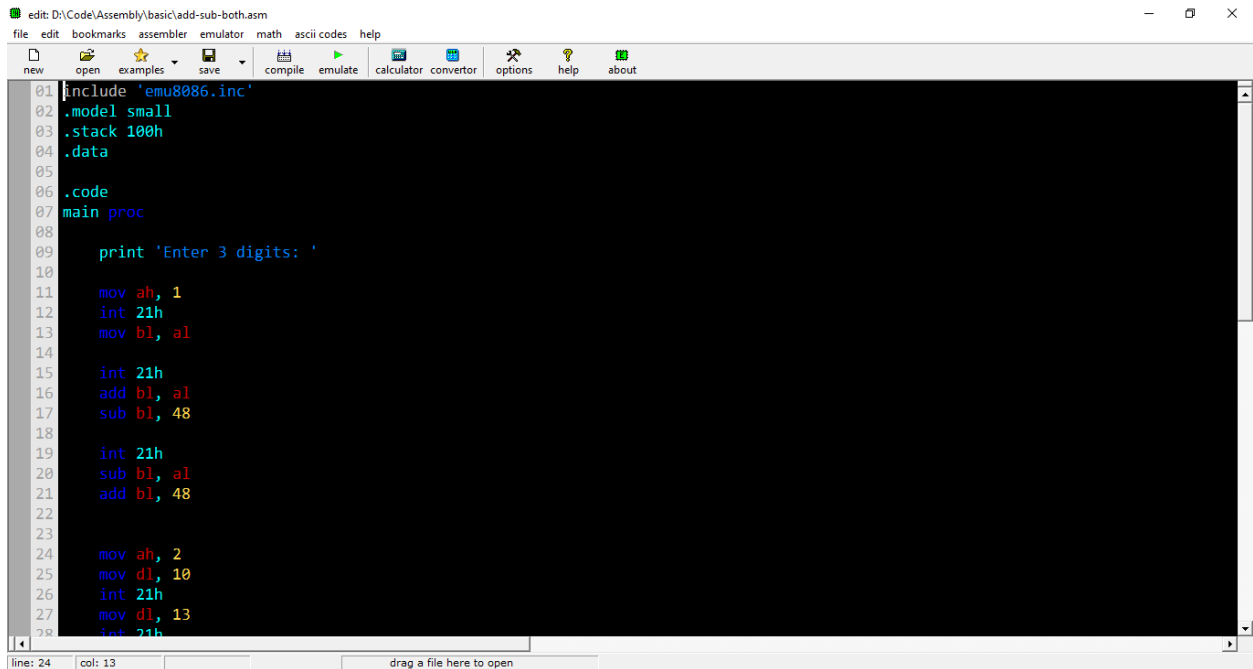


Taking input and printing output





Addition & Subtraction



```
edit: D:\Code\Assembly\basic\add-sub-both.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
13 mov bl, al
14
15 int 21h
16 add bl, al
17 sub bl, 48
18
19 int 21h
20 sub bl, al
21 add bl, 48
22
23
24 mov ah, 2
25 mov dl, 10
26 int 21h
27 mov dl, 13
28 int 21h
29
30
31 print 'A + B - C = '
32 mov dl, bl
33 int 21h
34
35
36 main endp
37 end main
```

line: 24 col: 13 drag a file here to open

edit: D:\Code\Assembly\basic\add-sub-both.asm

file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator convertor options help about

13 mov bl, al

14

15 int 21h

16 add bl, al

17 sub bl, 48

18

19 int 21h

20 sub bl, al

21 add bl, 48

22

23

24 mov ah, 2

25 mov dl, 10

26 int 21h

27 mov dl, 13

28 int 21h

29

30

31 print 'A + B - C = '

32 mov dl, bl

33 int 21h

34

35

36 main endp

37 end main

original source co...

```
24 mov ah, 2
25 mov dl, 10
26 int 21h
27 mov dl, 13
28 int 21h
29
30
31 print 'A + B - C = '
32 mov dl, bl
33 int 21h
```

emulator: add-sub-both.exe

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	02	33
BX	00	33
CX	01	6E
DX	00	33
CS	0720	
IP	0082	
SS	0710	
SP	0100	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

message

the emulator is halted

emulator screen (80x25 chars)

```
Enter 3 digits: 232
A + B - C = 3
```

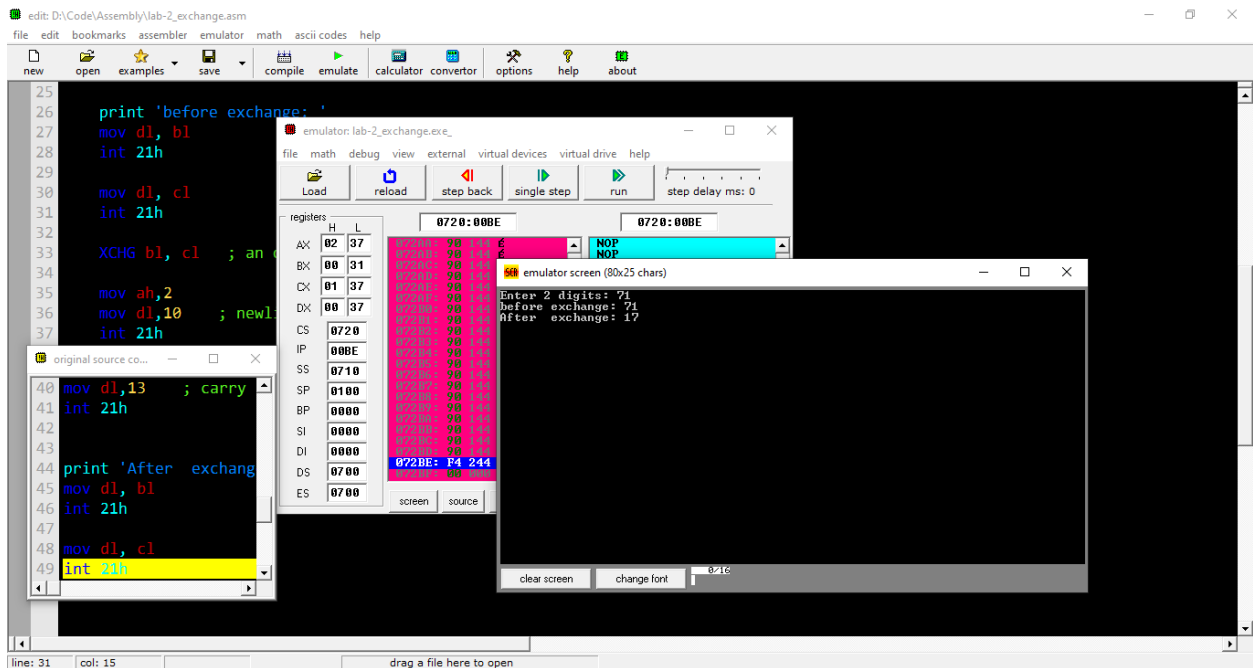
clear screen change font 8x16

line: 24 col: 13 drag a file here to open

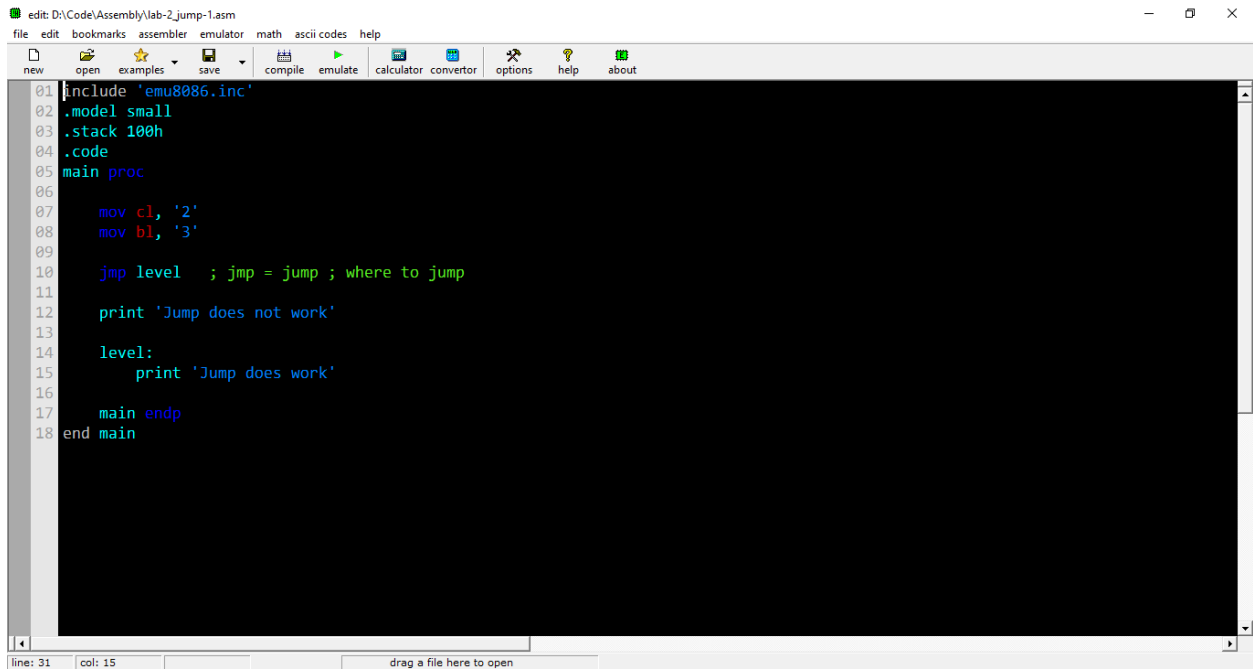
Exchange between 2 numbers

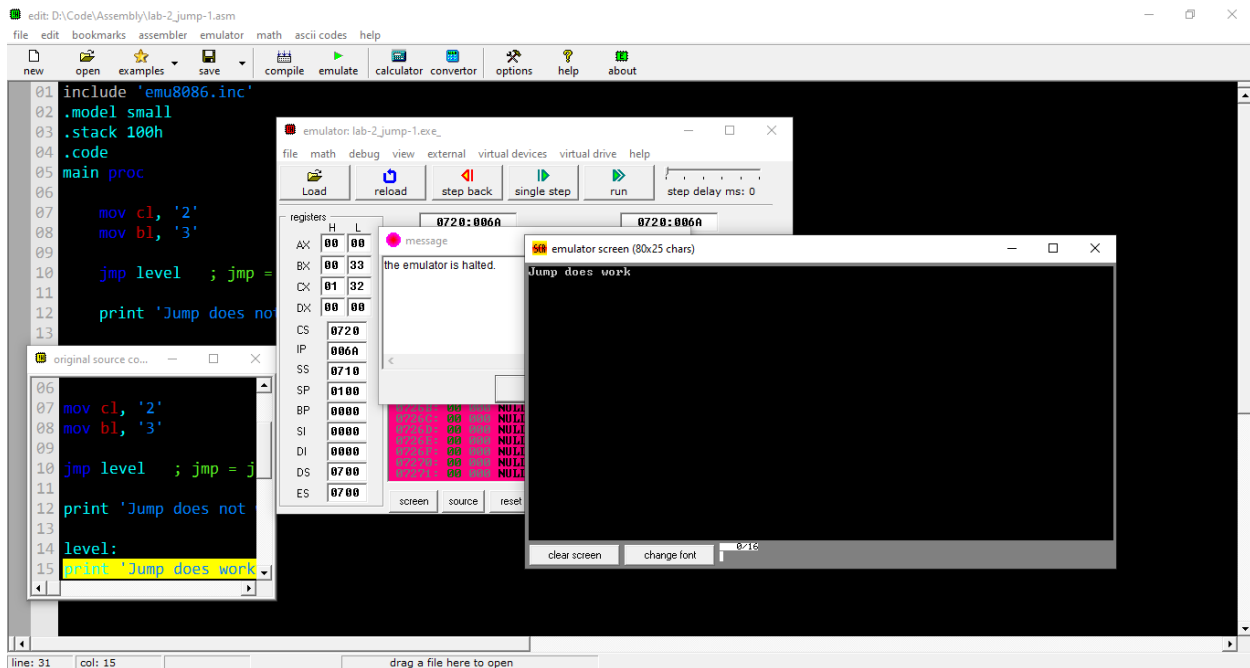
```
edit: D:\Code\Assembly\lab-2_exchange.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
01 include 'emu8086.inc'
02 .model small
03 .stack 100h
04 .data
05
06 .code
07 main proc
08
09     print 'Enter 2 digits: '
10
11     mov ah, 1
12     int 21h
13     mov bl, al
14
15     int 21h
16     mov cl, al
17
18
19     mov ah, 2
20     mov dl, 10 ; newline
21     int 21h
22     mov dl, 13 ; carry return
23     int 21h
24
25
26     print 'before exchange: '
27     mov dl, bl
28     int 21h
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

```
edit: D:\Code\Assembly\lab-2_exchange.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
25
26     print 'before exchange: '
27     mov dl, bl
28     int 21h
29
30     mov dl, cl
31     int 21h
32
33     XCHG bl, cl ; an off code for exchange
34
35     mov ah, 2
36     mov dl, 10 ; newline
37     int 21h
38     mov dl, 13 ; carry return
39     int 21h
40
41     print 'After exchange: '
42     mov dl, bl
43     int 21h
44
45     mov dl, cl
46     int 21h
47
48
49     main endp
50 end main
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

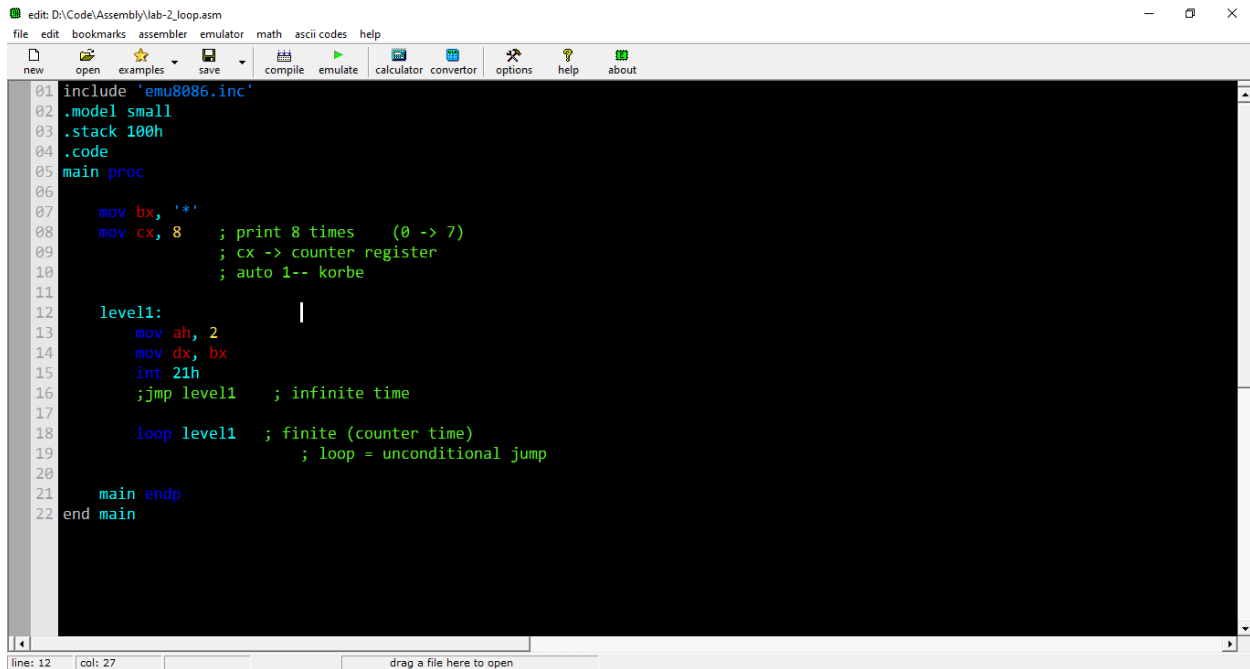


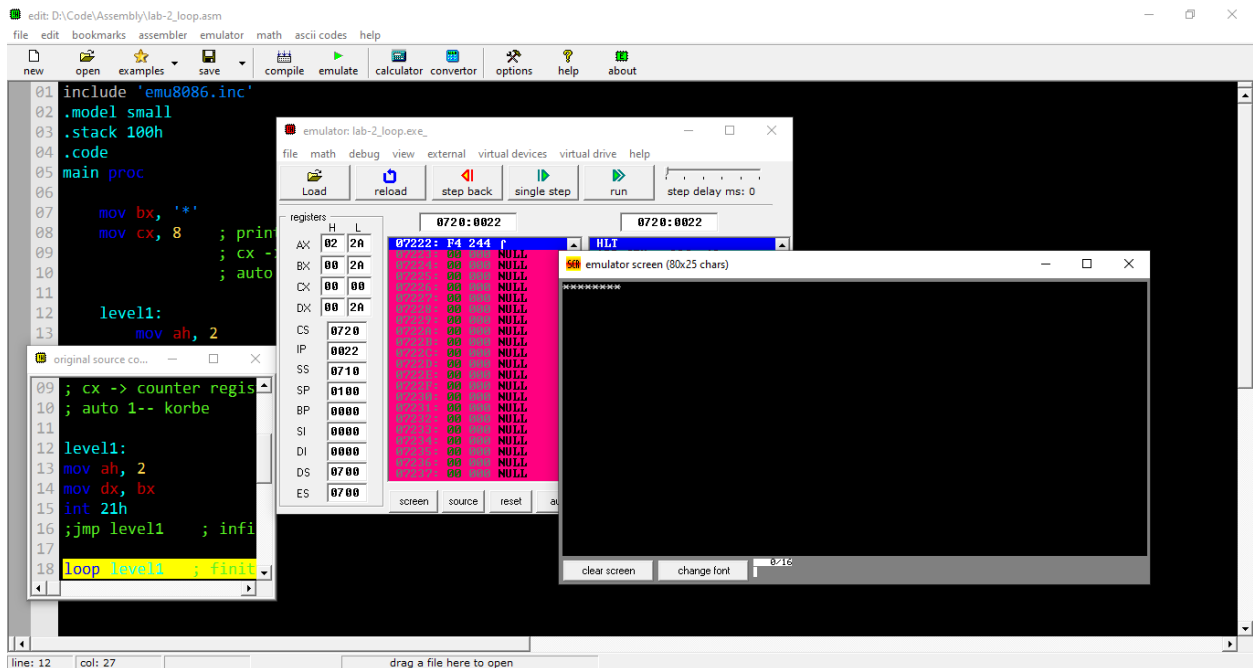
Use of 'jmp'



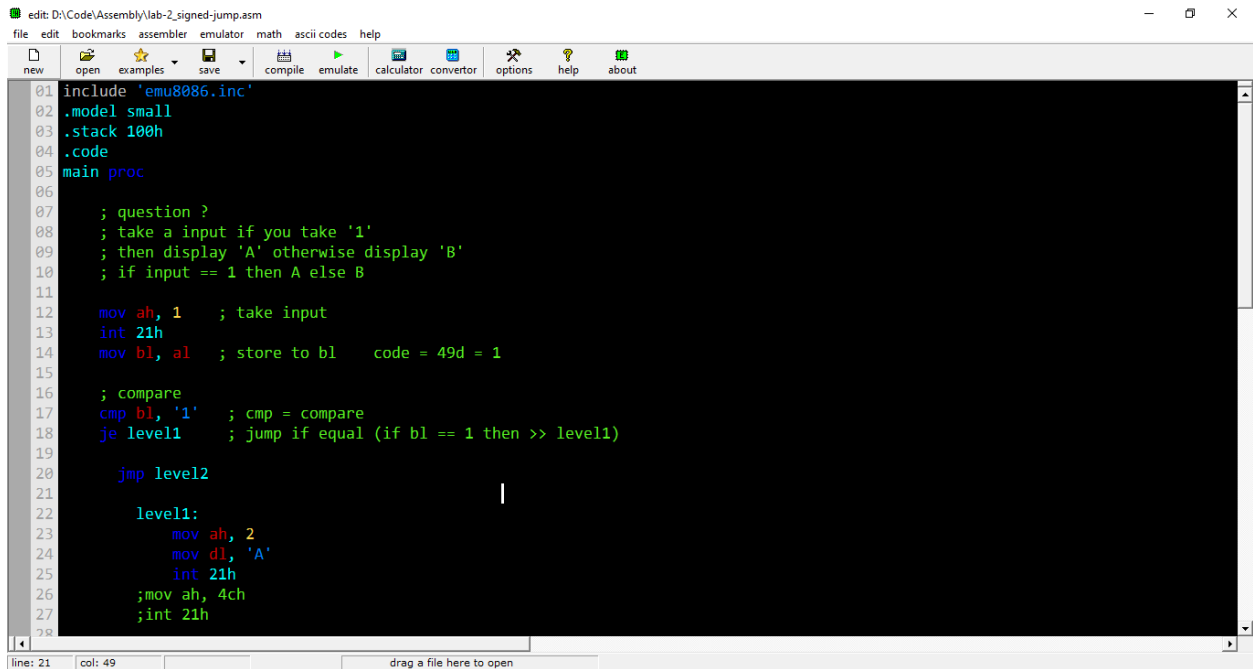


Print star(*) 8 times by for loop





Take an input. If you take '1' then display 'A' otherwise display 'B'



```

edit: D:\Code\Assembly\lab-2_signed-jump.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
16 ; compare
17 cmp bl, '1' ; cmp = compare
18 je level1 ; jump if equal (if bl == 1 then >> level1)
19
20 jmp level2
21
22 level1:
23 mov ah, 2
24 mov dl, 'A'
25 int 21h
26 ;mov ah, 4ch
27 ;int 21h
28
29 jmp exit
30
31 level2:
32 mov ah, 2
33 mov dl, 'B'
34 int 21h
35
36 exit:
37 mov ah, 4ch
38 int 21h
39
40 main endp
41 end main

```

line: 21 col: 49 drag a file here to open

```

edit: D:\Code\Assembly\lab-2_signed-jump.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
07 ; question ?
08 ; take a input if you take '1'
09 ; then display 'A'
10 ; if input == 1 then
11
12 mov ah, 1 ; take
13 int 21h
14 mov bl, al ; store
15
16 ; compare
17 cmp bl, '1' ; cmp
18 je level1 ; jump
19

```

emulator: lab-2_signed-jump.exe_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers H L

AX	00 41	FP	00 00	RES	00 00
BX	00 31	BP	00 00	CD	00 00
CX	01 1F	DI	00 00	CD	00 00
DX	00 41	SI	00 00	CD	00 00
CS	F4 00	SP	00 00	CD	00 00
IP	02 04	BP	00 00	CD	00 00
SS	07 10	SI	00 00	CD	00 00
SP	00 FA	DI	00 00	CD	00 00
BP	00 00	SI	00 00	CD	00 00
SI	00 00	DI	00 00	CD	00 00
DI	00 00	SI	00 00	CD	00 00
DS	07 00	DI	00 00	CD	00 00
ES	07 00	DI	00 00	CD	00 00

screen source reset

emulator screen (80x25 chars)

clear screen change font 8x16

original source co...

```

29 jmp exit
30
31 level2:
32 mov ah, 2
33 mov dl, 'B'
34 int 21h
35
36 exit:
37 mov ah, 4ch
38 int 21h
39

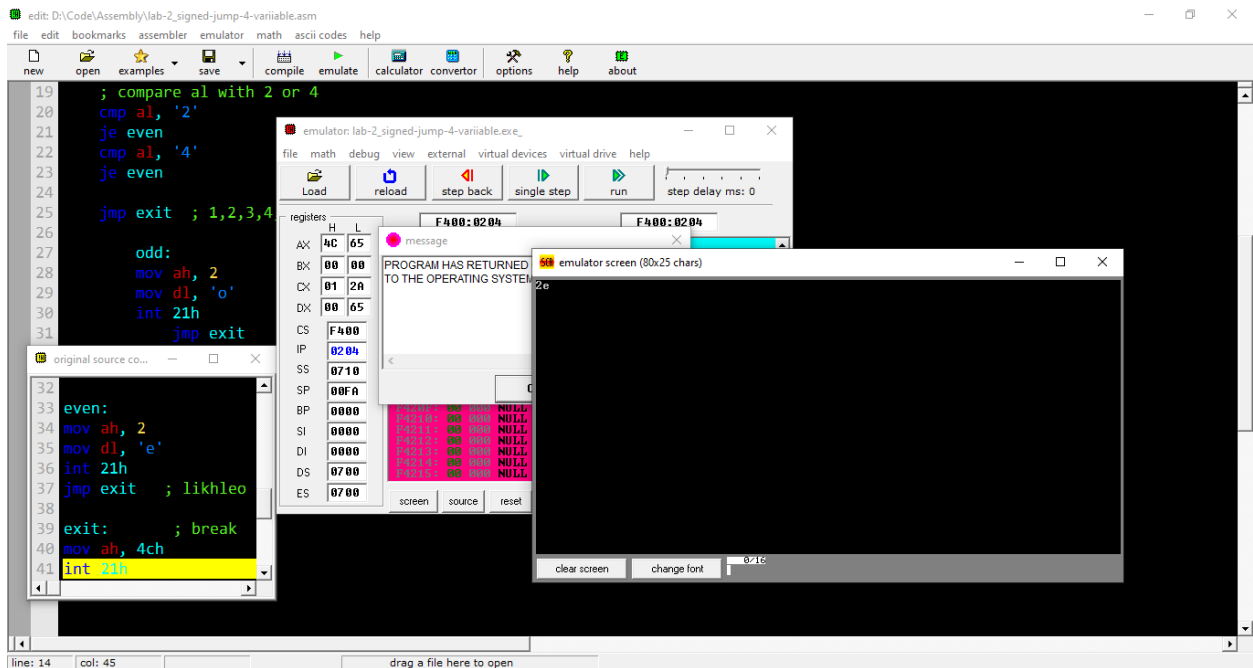
```

line: 21 col: 49 drag a file here to open

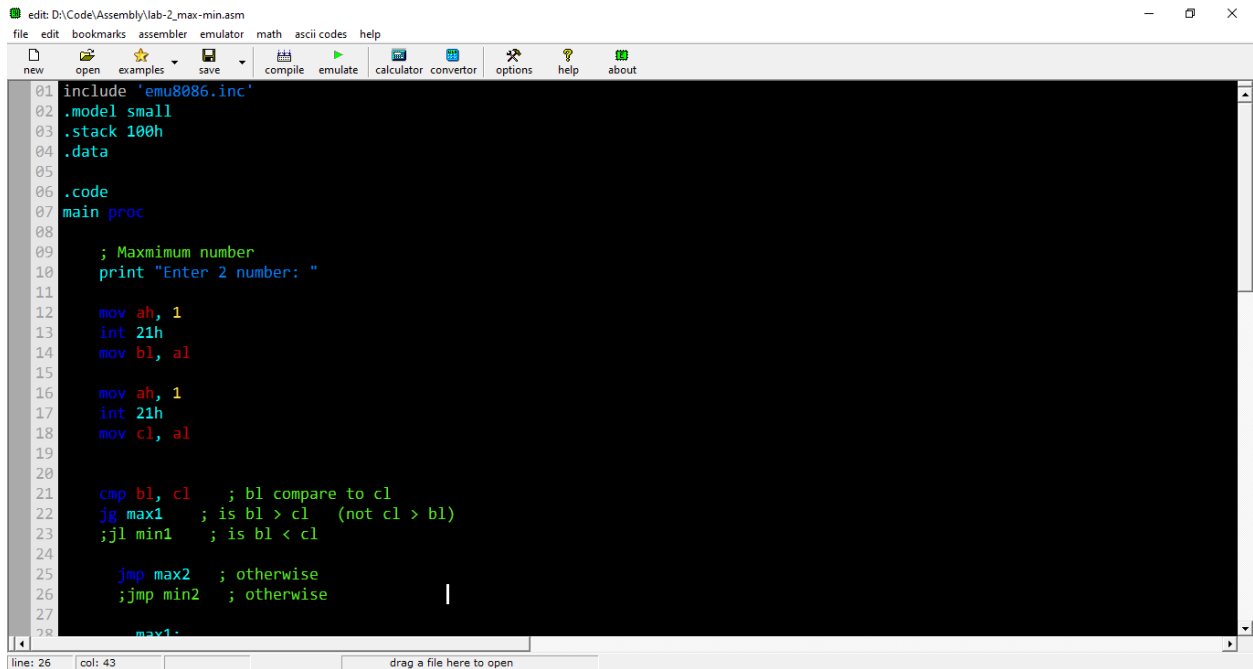
If input is 1 or 3 then print 'o' or if the input is 2 or 4 then print 'e'

```
edit: D:\Code\Assembly\lab-2_signed-jump-4-variable.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
01 .model small
02 .stack 100h
03 .code
04 main proc
05
06 ; question ?
07 ; if (al == 1 || 3) >> 'o'
08 ; if (al == 2 || 4) >> 'e'
09
10 mov ah, 1
11 int 21h
12
13 ; compare al with 1 or 3
14 cmp al, '1'
15 je odd
16 cmp al, '3'
17 je odd
18
19 ; compare al with 2 or 4
20 cmp al, '2'
21 je even
22 cmp al, '4'
23 je even
24
25 jmp exit ; 1,2,3,4, chara onno kisu dileo execute korb
26
27 odd:
28 mov ah, 2
29 int 21h
30 jmp exit
31
32 even:
33 mov ah, 2
34 mov dl, 'e'
35 int 21h
36 jmp exit ; likhleo same, na likhleo same
37
38 exit: ; break
39 mov ah, 4ch
40 int 21h
41
42
43 main endp
44 end main
```

```
edit: D:\Code\Assembly\lab-2_signed-jump-4-variable.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
19 ; compare al with 2 or 4
20 cmp al, '2'
21 je even
22 cmp al, '4'
23 je even
24
25 jmp exit ; 1,2,3,4, chara onno kisu dileo execute korb
26
27 odd:
28 mov ah, 2
29 mov dl, 'o'
30 int 21h
31 jmp exit
32
33 even:
34 mov ah, 2
35 mov dl, 'e'
36 int 21h
37 jmp exit ; likhleo same, na likhleo same
38
39 exit: ; break
40 mov ah, 4ch
41 int 21h
42
43 main endp
44 end main
```



Find maximum/minimum between 2 numbers



```
edit: D:\Code\Assembly\lab-2_max-min.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
22 jg max1 ; is b1 > c1 (not c1 > b1)
23 jnl min1 ; is b1 < c1
24
25 jmp max2 ; otherwise
26 jmp min2 ; otherwise
27
28 max1:
29 print "Maximum number : "
30 mov ah, 2
31 mov dl, b1 ; if b1 is greater than c1
32 int 21h
33
34
35 max2:
36 print "Maximum number : "
37 mov ah, 2
38 mov dl, c1 ; if c1 is greater than b1
39 int 21h
40
41
42 exit: ; break
43 mov ah, 4ch
44 int 21h
45
46 main endp
47 end main
```

line: 26 col: 43 drag a file here to open

edit: D:\Code\Assembly\lab-2_max-min.asm
file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator convertor options help about

22 jg max1 ; is b1 > c1 (not c1 > b1)
23 jnl min1 ; is b1 < c1
24
25 jmp max2 ; otherwise
26 jmp min2 ; otherwise
27
28 max1:
29 print "Maximum number : "
30 mov ah, 2
31 mov dl, b1 ; if b1 is greater than c1
32 int 21h
33
34
35 max2:
36 print "Maximum number : "
37 mov ah, 2
38 mov dl, c1 ; if c1 is greater than b1
39 int 21h
40
41
42 exit: ; break
43 mov ah, 4ch
44 int 21h

original source co...

```
35 max2:
36 print "Maximum number : "
37 mov ah, 2
38 mov dl, c1 ; if c1
39 int 21h
40
41
42 exit: ; break
43 mov ah, 4ch
44 int 21h
```

emulator: lab-2_max-min.exe

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00 36	
BX	00 34	
CX	01 36	
DX	00 36	
CS	F4 00	
IP	02 04	
SS	07 10	
SP	0B F8	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

emulator screen (80x25 chars)

Enter 2 number: 46Maximum number :6

clear screen change font 8*16

Convert small to capital letter

```
edit: D:\Code\Assembly\lab-3_small-to-capital.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
01 include 'emu8086.inc'
02 .model small
03 .stack 100h
04 .data
05
06 .code
07 main proc
08
09     print 'Enter (a-z) = '
10     mov ah, 1
11     int 21h
12     mov bl, al
13
14     sub bl, 32    ; small to capital difference
15     ;add bl, 32    ; capital to small difference
16
17     mov ah, 2
18     mov dl, 10
19     int 21h
20     mov dl, 13
21     int 21h
22
23     print 'Capital form: '
24     mov dl, bl
25     int 21h
26
27     main endp
28 end main
line: 20 col: 33 drag a file here to open
```

edit: D:\Code\Assembly\lab-3_small-to-capital.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about

emulator: lab-3_small-to-capital.exe_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers H L

AX	02	53
BX	00	53
CX	01	63
DX	00	53
CS	0720	
IP	0077	
SS	0710	
SP	0100	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

message

the emulator is halted.

emulator screen (80x25 chars)

```
Enter (a-a) = s
Capital form: S
```

original source code...

```
16 mov ah, 2
17 mov dl, 10
18 int 21h
19 int 21h
20 mov dl, 13
21 int 21h
22
23 print 'Capital form: '
24 mov dl, bl
25 int 21h
26
27 main endp
28 end main
line: 20 col: 33 drag a file here to open
```

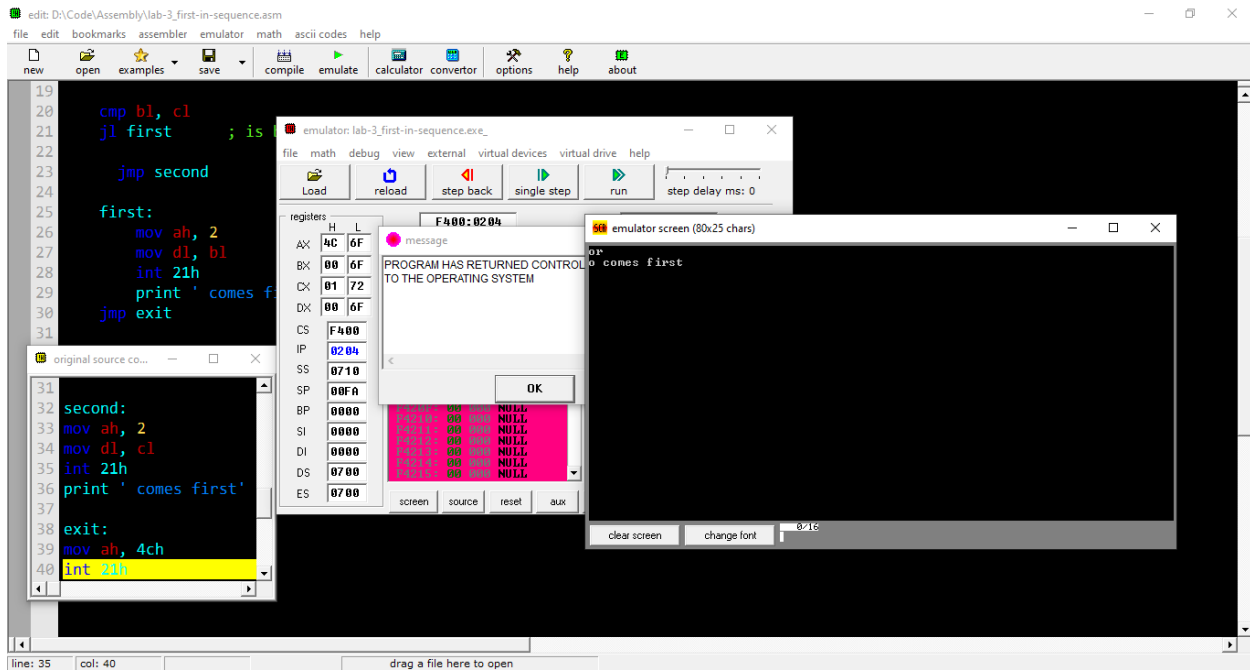
Show which value comes first in sequence between two

```
edit: D:\Code\Assembly\lab-3_first-in-sequence.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
01 include 'emu8086.inc'
02 .model small
03 .stack 100h
04 .code
05 main proc
06
07     mov ah, 1
08     int 21h
09     mov bl, al
10
11     int 21h
12     mov cl, al
13
14     mov ah, 2
15     mov dl, 10
16     int 21h
17     mov dl, 13
18     int 21h
19
20     cmp bl, cl
21     jl first      ; is bl < cl
22
23     jmp second
24
25 first:
26     mov ah, 2
27     mov dl, bl
28     int 21h
29
```

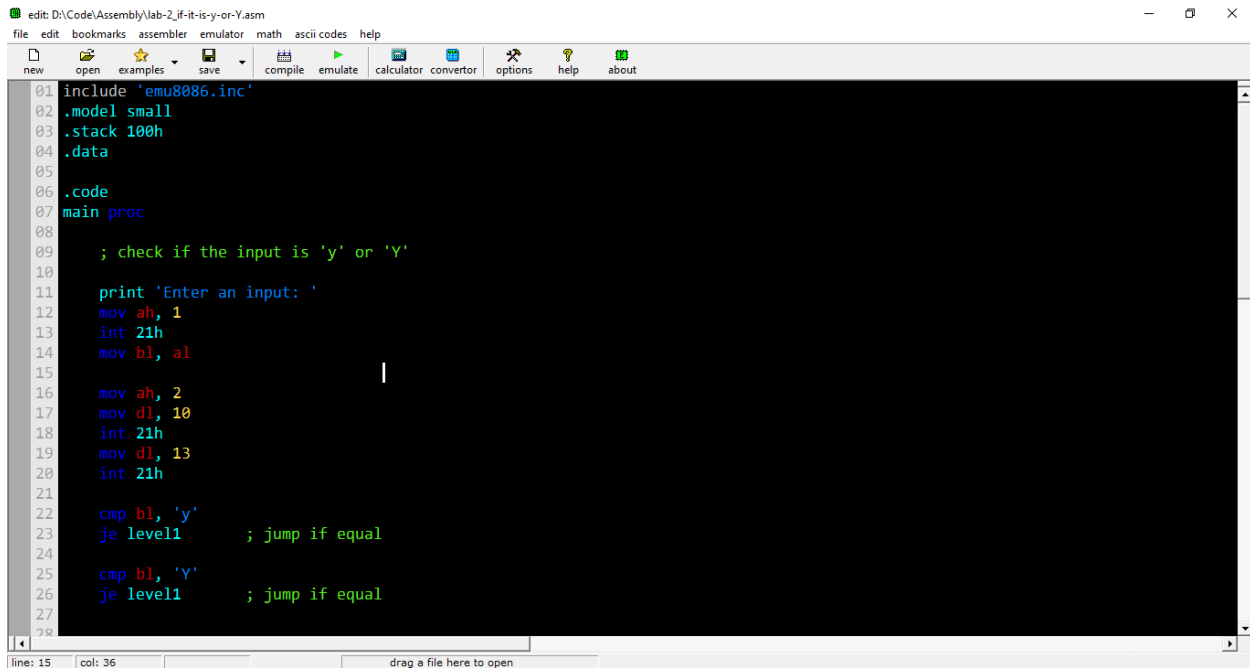
line: 20 col: 33 drag a file here to open

```
edit: D:\Code\Assembly\lab-3_first-in-sequence.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
19
20     cmp bl, cl
21     jl first      ; is bl < cl
22
23     jmp second
24
25 first:
26     mov ah, 2
27     mov dl, bl
28     int 21h
29     print ' comes first'
30     jmp exit
31
32 second:
33     mov ah, 2
34     mov dl, cl
35     int 21h
36     print ' comes first'
37
38 exit:
39     mov ah, 4ch
40     int 21h
41
42     main endp
43 end main
```

line: 17 col: 42 drag a file here to open



Check if the input is 'Y' or 'y'




```
edit: D:\Code\Assembly\lab-2_if-it-is-y-or-Y.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
22 cmp bl, 'y'
23 je level1 ; jump if equal
24
25 cmp bl, 'Y'
26 je level1 ; jump if equal
27
28
29 print 'Input is not y or Y'
30 jmp exit
31
32 level1:
33 print 'The value is '
34 mov ah, 2
35 mov dl, bl
36 int 21h
37 jmp exit
38
39 exit:
40 mov ah, 4
41 int 21h
42
43 main endp
44 end main
```

line: 15 col: 36 drag a file here to open

edit: D:\Code\Assembly\lab-2_if-it-is-y-or-Y.asm
file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator convertor options help about

emulator: lab-2_if-it-is-y-or-Y.exe

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	04	79
BX	00	79
CX	01	A0
DX	00	79
CS	F4	00
IP	02	04
SS	07	10
SP	0B	F8
BP	00	00
SI	00	00
DI	00	00
DS	07	00
ES	07	00

message

INT 21h, AH=04h - not supported yet
refer to the list of supported interrupts.

OK

emulator screen (80x25 chars)

Enter an input: y
The value is y

original source code

```
32 level1:
33 print 'The value is '
34 mov ah, 2
35 mov dl, bl
36 int 21h
37 jmp exit
38
39 exit:
40 mov ah, 4
41 int 21h
```

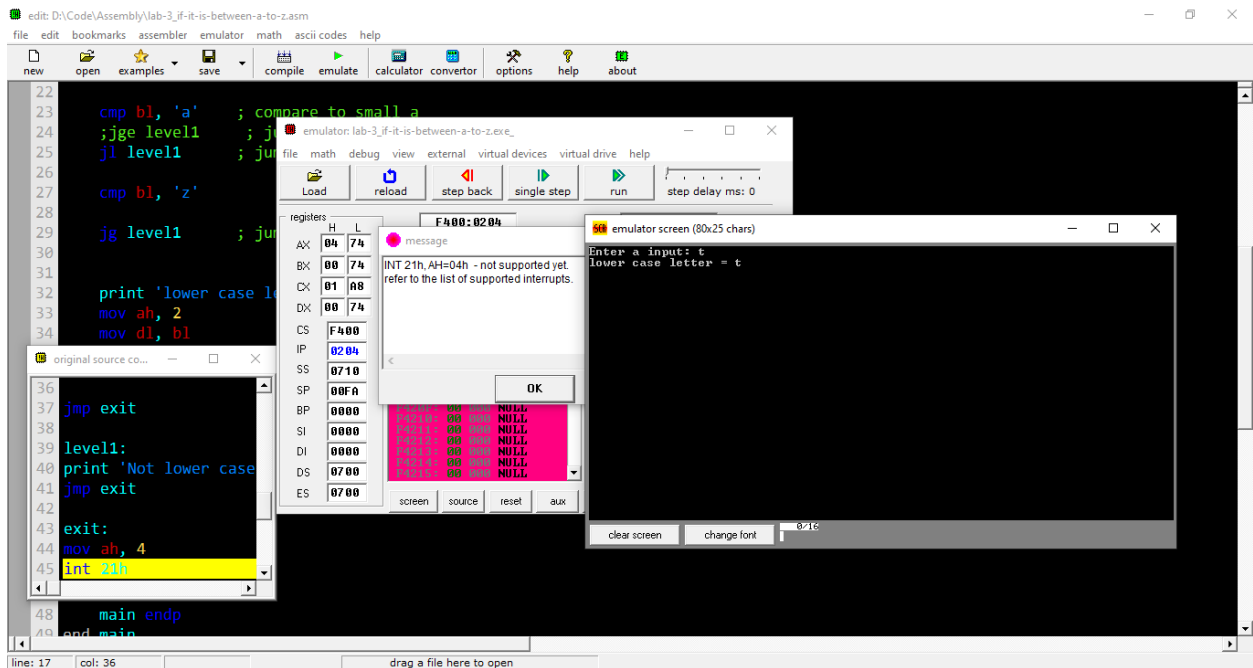
clear screen change font 8x16

line: 15 col: 36 drag a file here to open

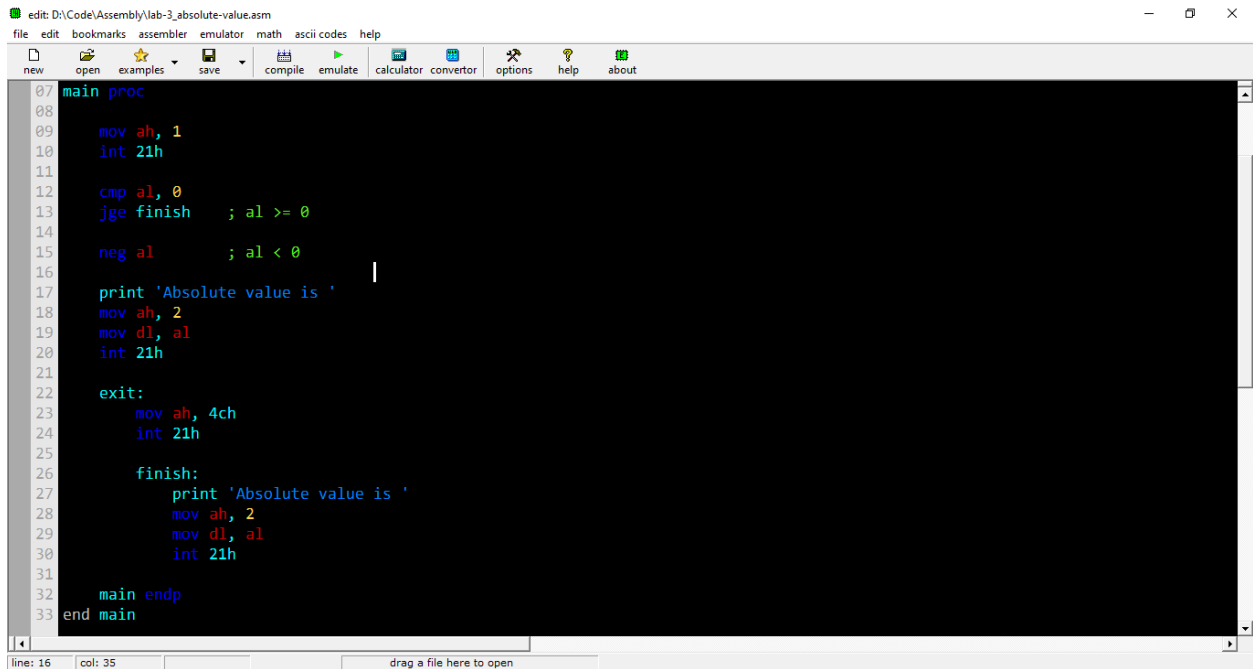
Checking small letter alphabet

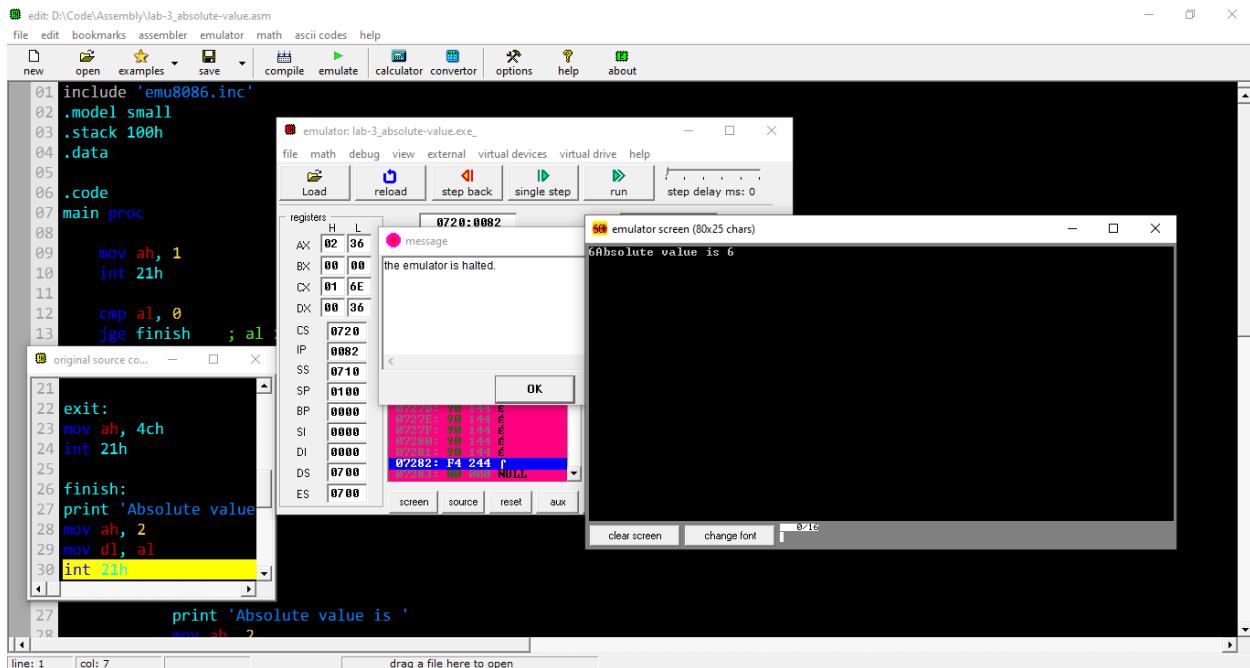
```
edit: D:\Code\Assembly\lab-3_if-it-is-between-a-to-z.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
01 include 'emu8086.inc'
02 .model small
03 .stack 100h
04 .data
05
06 .code
07 main proc
08
09     ; check if the input is between 'a' to 'z' ?    97 - 122
10     ; input >= a  and input <= z
11
12     print 'Enter a input: '
13     mov ah, 1
14     int 21h
15     mov bl, al
16
17     mov ah, 2
18     mov dl, 10
19     int 21h
20     mov dl, 13
21     int 21h
22
23     cmp bl, 'a'    ; compare to small a
24     ;jge level1    ; jump if greater and equal
25     jl level1     ; jump if less than    <a
26
27     cmp bl, 'z'
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49 end main
line: 17 col: 36 drag a file here to open
```

```
edit: D:\Code\Assembly\lab-3_if-it-is-between-a-to-z.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
22
23     cmp bl, 'a'    ; compare to small a
24     ;jge level1    ; jump if greater and equal
25     jl level1     ; jump if less than    <a
26
27     cmp bl, 'z'
28
29     jg level1     ; jump if greater    >z
30
31
32     print 'lower case letter = '    ; so a=> _ <=z
33     mov ah, 2
34     mov dl, bl
35     int 21h
36
37     jmp exit
38
39     level1:
40         print 'Not lower case letter'
41         jmp exit
42
43     exit:
44         mov ah, 4
45         int 21h
46
47
48     main endp
49 end main
line: 17 col: 36 drag a file here to open
```

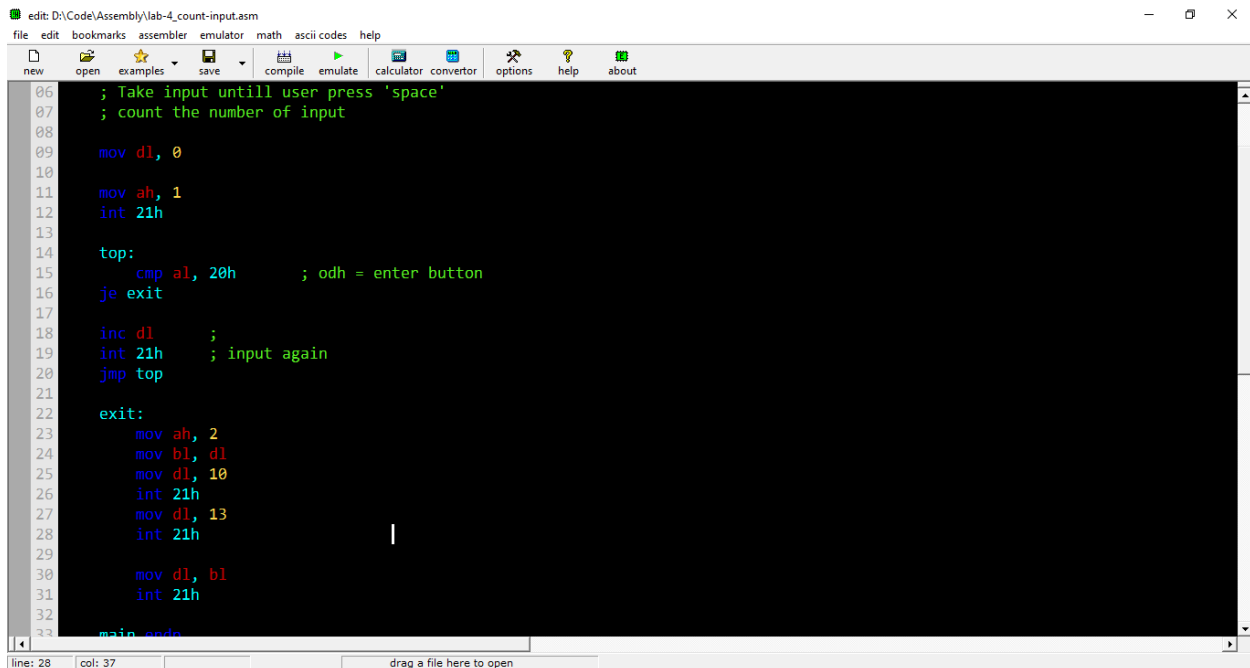


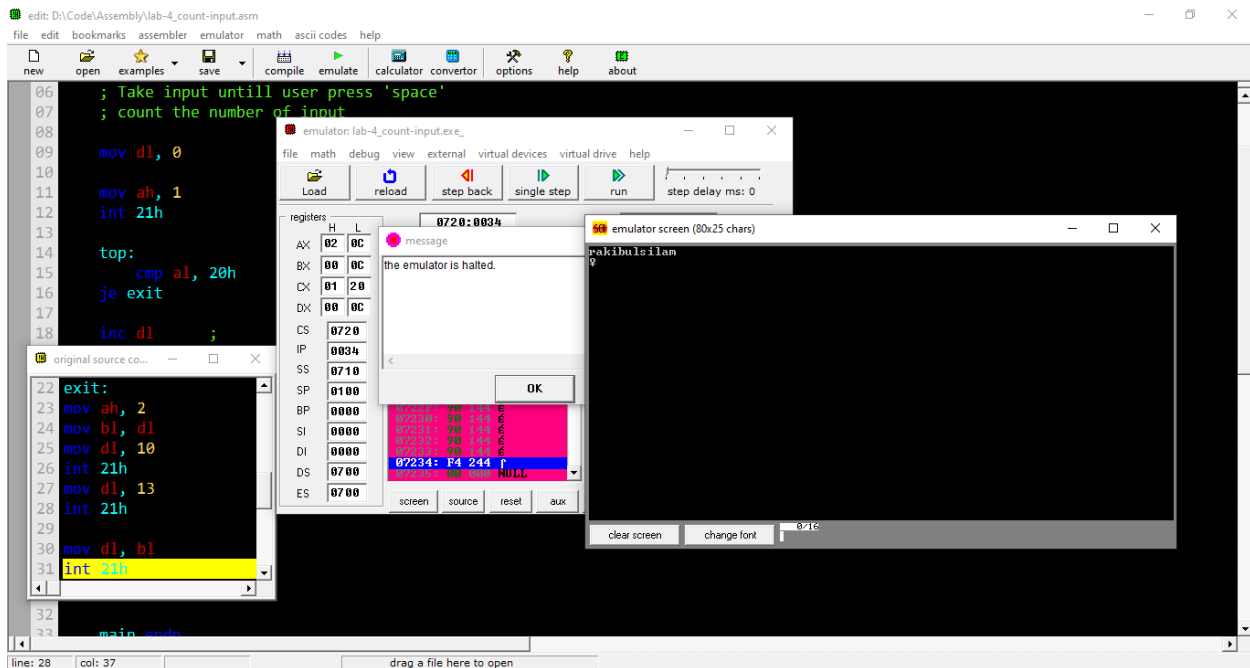
Find the absolute value



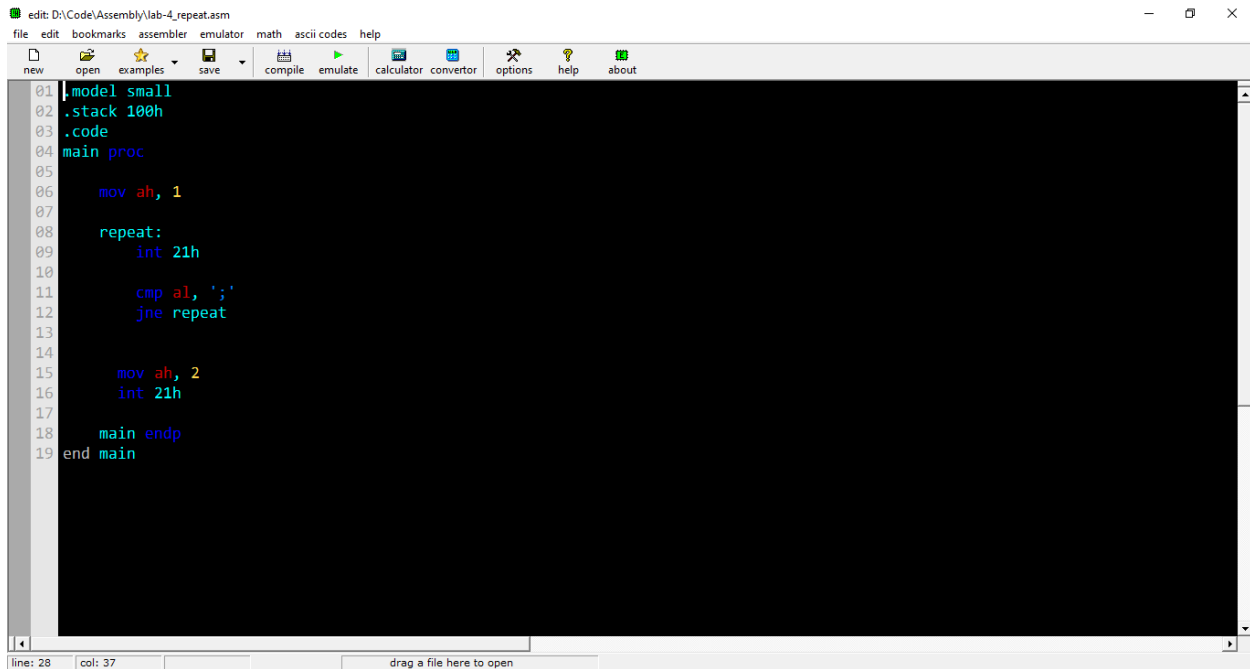


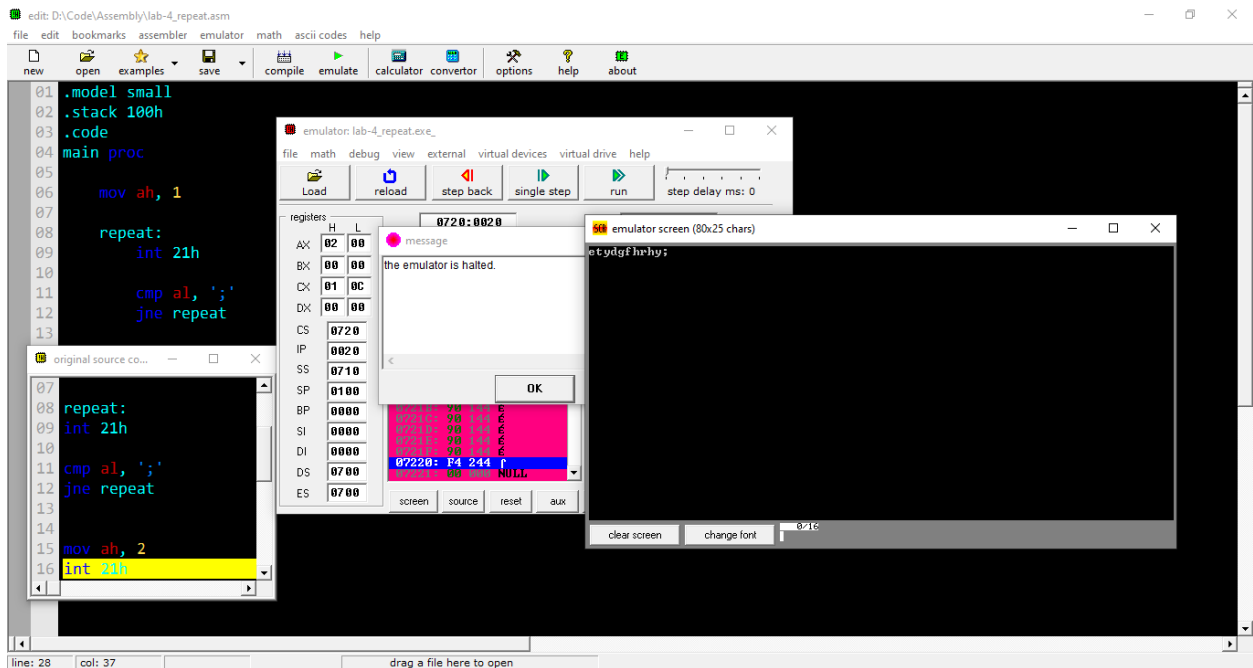
Count input until the space is given



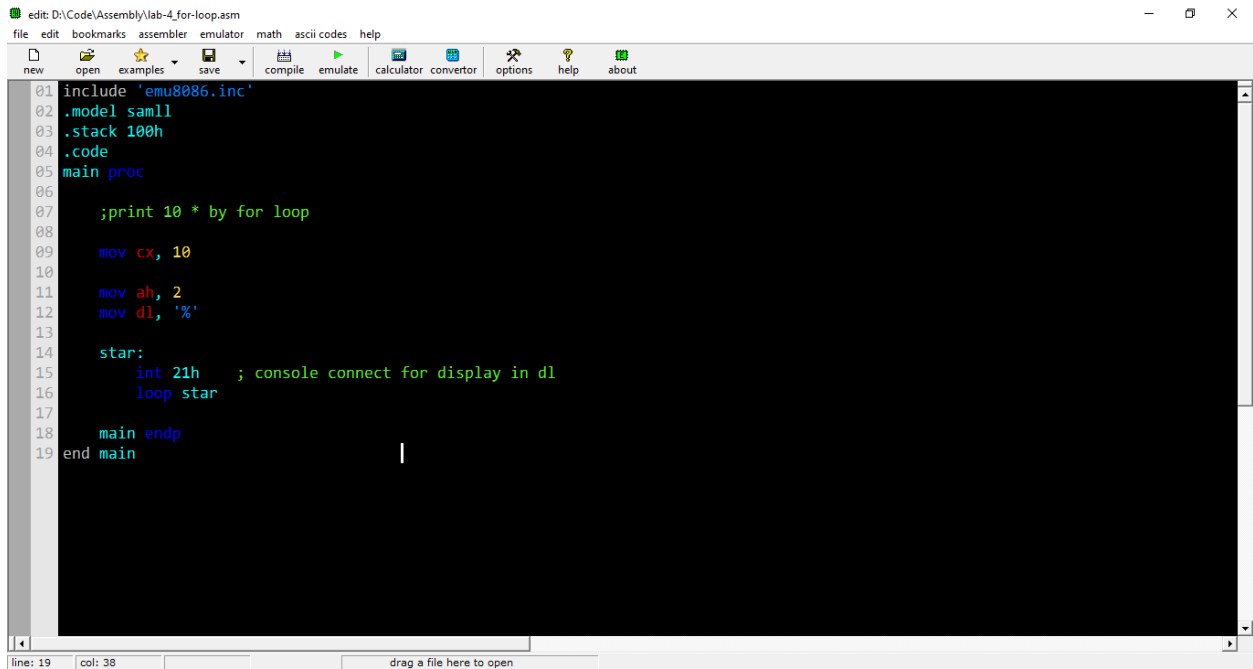


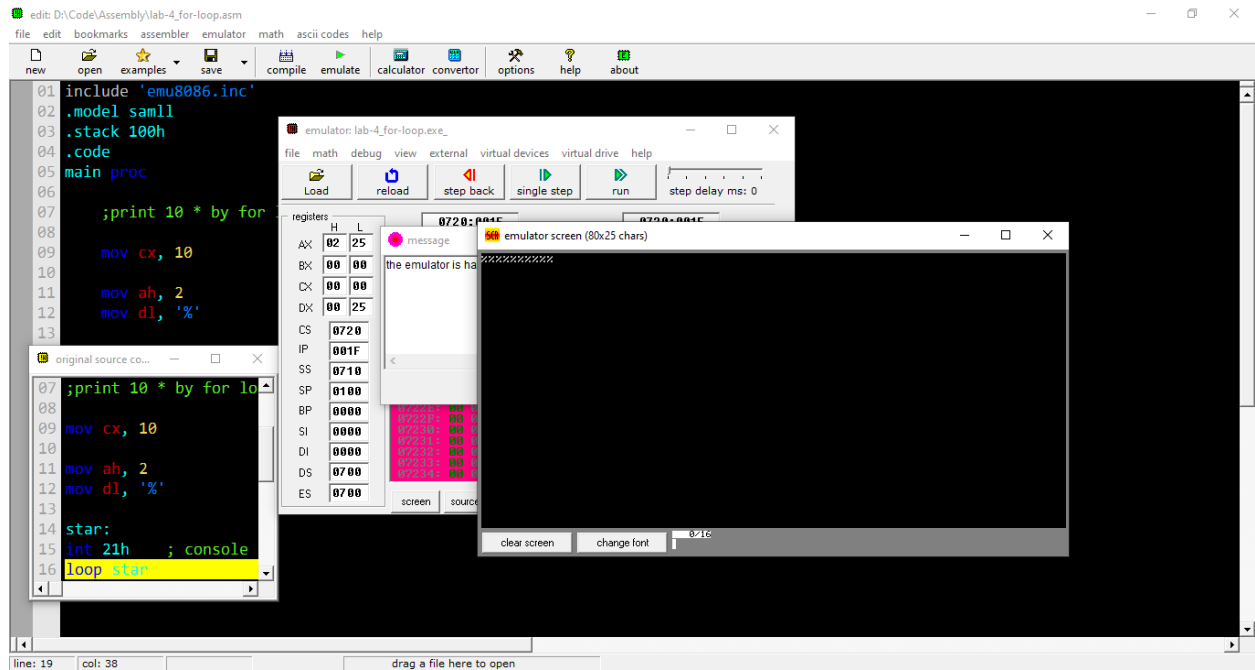
Use of repeat



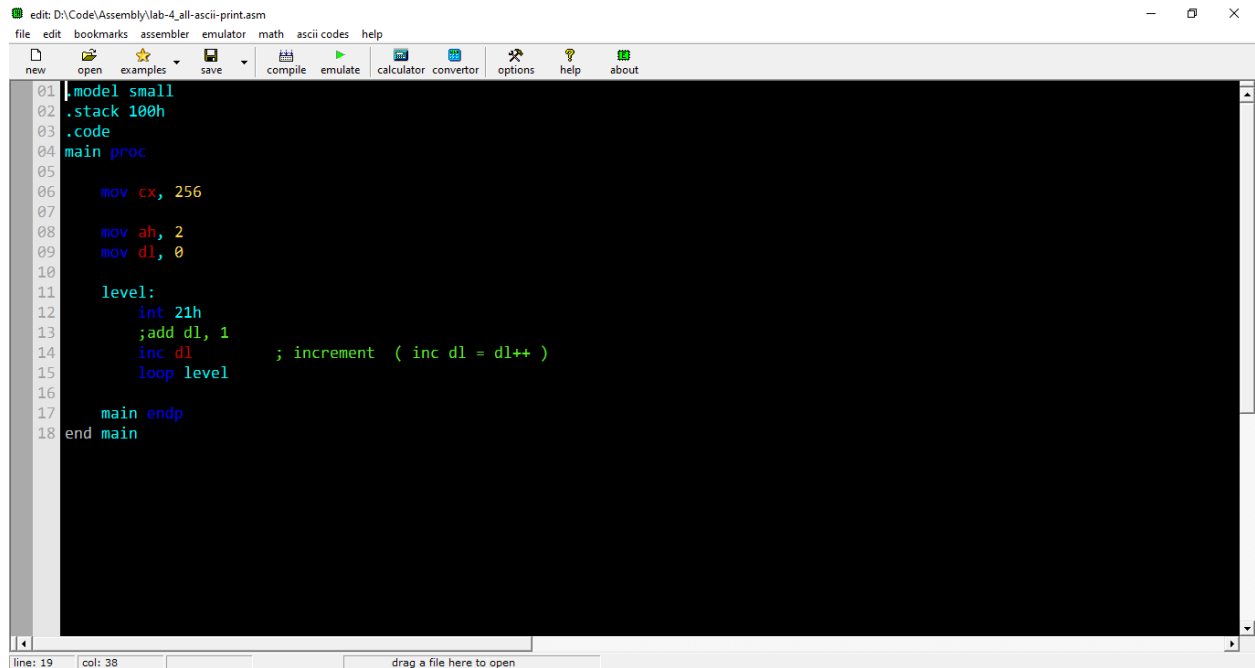


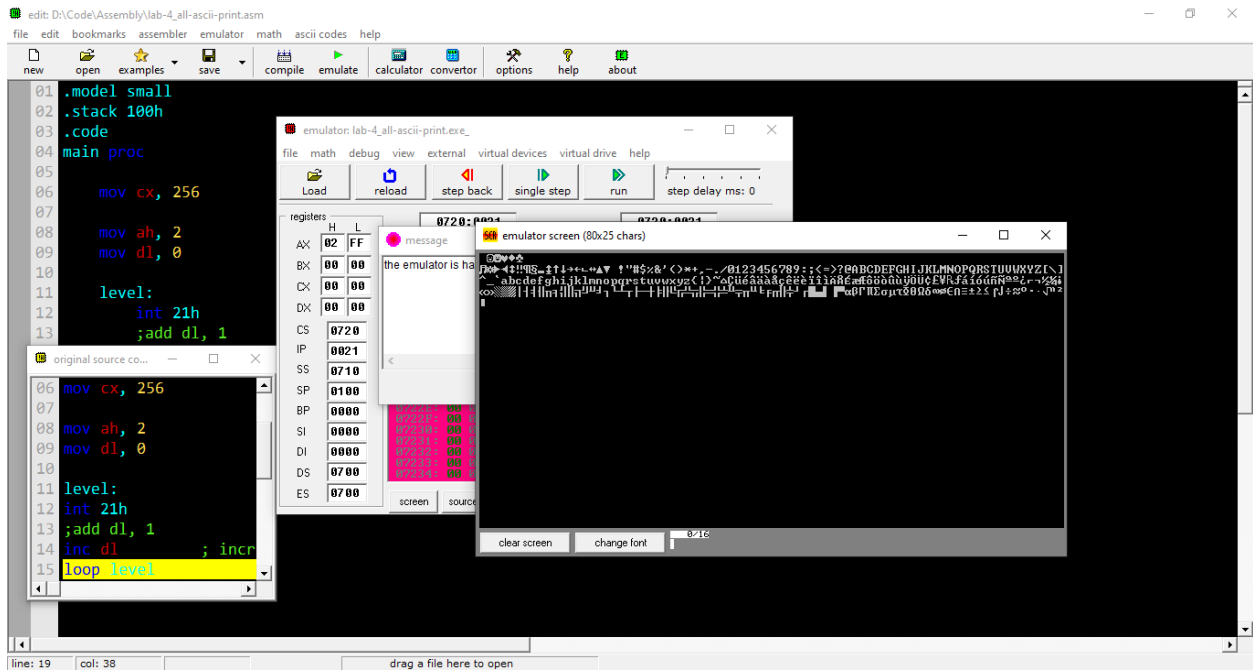
Print 10 '%' by using for loop



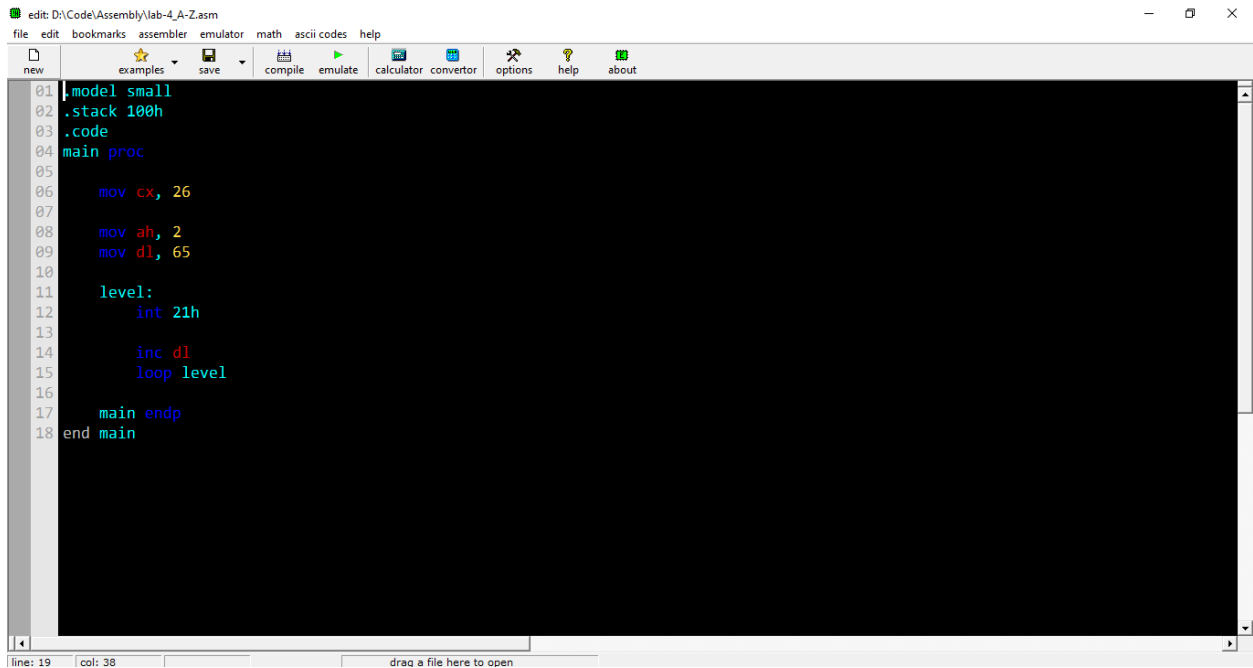


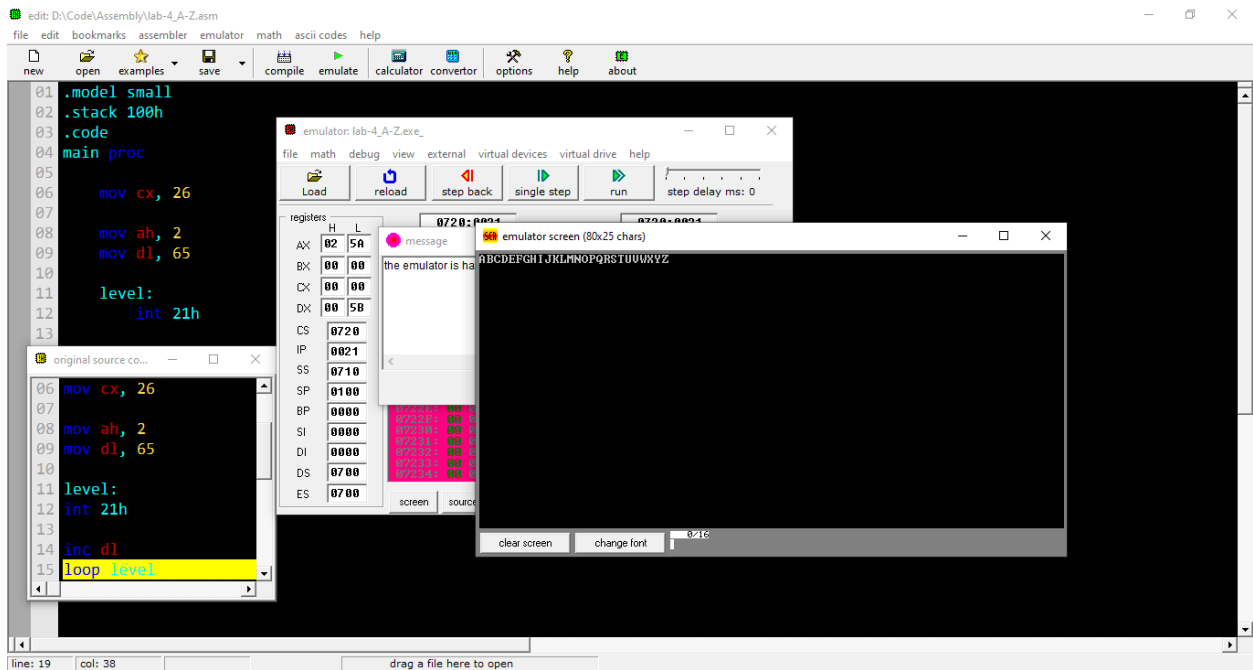
Printing all ascii values



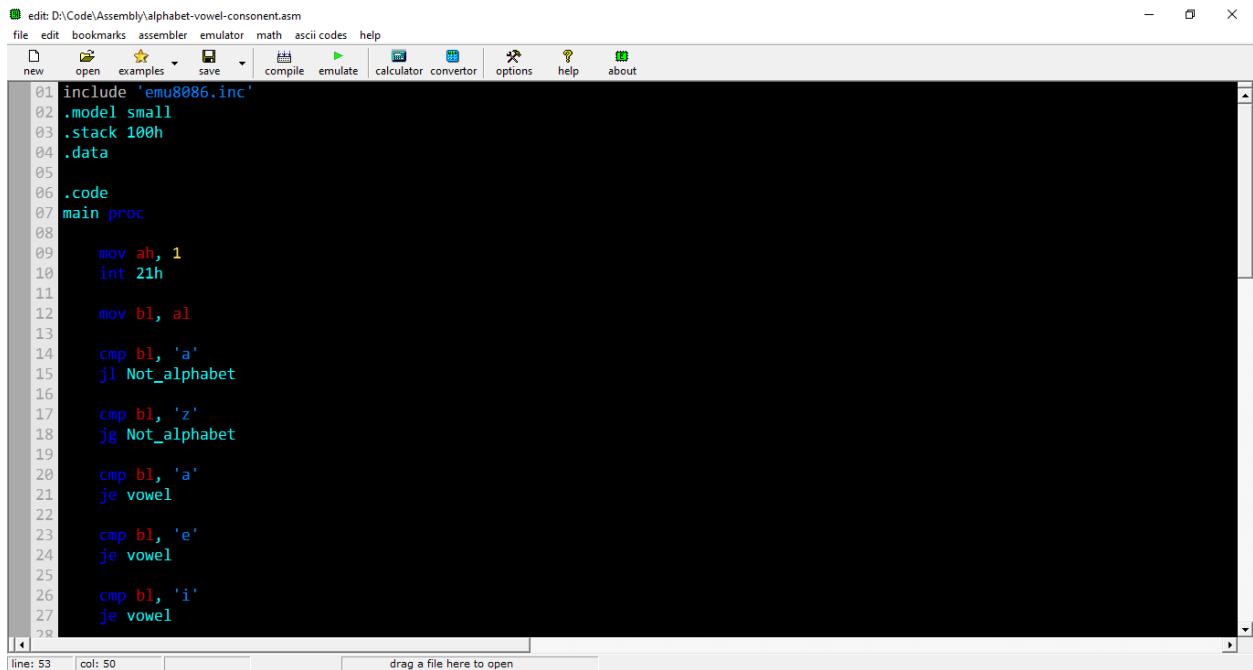


Print capital letters





Check an input is alphabet or vowel or anything else



```
edit: D:\Code\Assembly\alphabet-vowel-consonant.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
25
26 cmp bl, 'i'
27 je vowel
28
29 cmp bl, 'o'
30 je vowel
31
32 cmp bl, 'u'
33 je vowel
34
35
36 print ' It is an alphabet but not a vowel'
37 jmp exit
38
39
40 vowel:
41 print ' It is an alphabet and a vowel too'
42 jmp exit
43
44
45 Not_alphabet:
46 print ' Not an alphabet'
47
48 exit:
49 mov ah, 4
50 int 21h
51
52 main_code
line: 53 col: 50 drag a file here to open
```

```
edit: D:\Code\Assembly\alphabet-vowel-consonant.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
37 jmp exit
38
39
40 vowel:
41 print ' It is an alphabet and a vowel too'
42 jmp exit
43
44
45 Not_alphabet:
46 print ' Not an alphabet'
47
48 exit:
49 mov ah, 4
50 int 21h
line: 18 col: 49 drag a file here to open
```

emulator: alphabet-vowel-consonant.exe

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers H L

AX	04	73
BX	00	73
CX	01	03
DX	00	00
CS	F4	00
IP	02	04
SS	07	10
SP	00	F8
BP	00	00
SI	00	00
DI	00	00
DS	07	00
ES	07	00

message

emulator screen (80x25 chars)

INT 21h, AH=04h
refer to the list of s

original source co...

```
41 print ' It is an alph
42 jmp exit
43
44
45 Not_alphabet:
46 print ' Not an alphab
47
48 exit:
49 mov ah, 4
50 int 21h
```

clear screen change font 0.216

Take some inputs and show the outputs by stack

```
edit: D:\Code\Assembly\lab-5_stack.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about

06 main proc
07
08     mov cx, 0
09
10     mov ah, 1
11     int 21h
12
13     compare:
14         cmp al, 0dh
15         je output
16         inc cx
17         push ax
18         int 21h
19         jmp compare
20
21     output:
22         mov ah, 2
23         mov dl, 0ah
24         int 21h
25         mov dl, 0dh
26         int 21h
27
28     top:
29         pop dx
30         int 21h
31         loop top
32
33 main endp
```

line: 22 col: 24 drag a file here to open

edit: D:\Code\Assembly\lab-5_stack.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about

06 main proc
07
08 mov cx, 0
09
10 mov ah, 1
11 int 21h
12
13 compare:
14 cmp al, 0dh
15 je output
16 inc cx
17 push ax
18 int 21h
19
20
21
22
23
24
25
26
27
28 top:
29 pop dx
30 int 21h
31 loop top
32
33 main endp

original source co...
22 mov ah, 2
23 mov dl, 0ah
24 int 21h
25 mov dl, 0dh
26 int 21h
27
28 top:
29 pop dx
30 int 21h
31 loop top
32
33 main endp

emulator: lab-5_stack.exe
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0

registers
AX: 02 73
BX: 00 00
CX: 00 00
DX: 01 73
CS: 0720
IP: 0034
SS: 0710
SP: 0100
BP: 0000
SI: 0000
DI: 0000
DS: 0700
ES: 0700

message
the emulator is h...

emulator screen (80x25 chars)
shanto
otnahs

clear screen change font 8x16

line: 22 col: 24 drag a file here to open

Reversing string by push and pop

edit: D:\Code\Assembly\lab-5_string_reverse.asm

file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator convertor options help about

```
01 |.model small
02 |.stack 100h
03 |.data
04 |.code
05 |
06 |main proc
07 |
08 |    xor cx, cx    ; ??
09 |
10 |    mov ah, 1
11 |    int 21h
12 |
13 |    compare:
14 |    cmp al, 0dh
15 |    je output
16 |    inc cx
17 |    push ax
18 |    int 21h
19 |    jmp compare
20 |
21 |
22 |    output:
23 |    jcxz exit:    ; JCXZ tests the contents of the CX register for 0
24 |    mov ah, 2
25 |    mov dl,10
26 |    int 21h
27 |    mov dl,13
28 |    int 21h
29 |
```

line: 21 col: 115 drag a file here to open

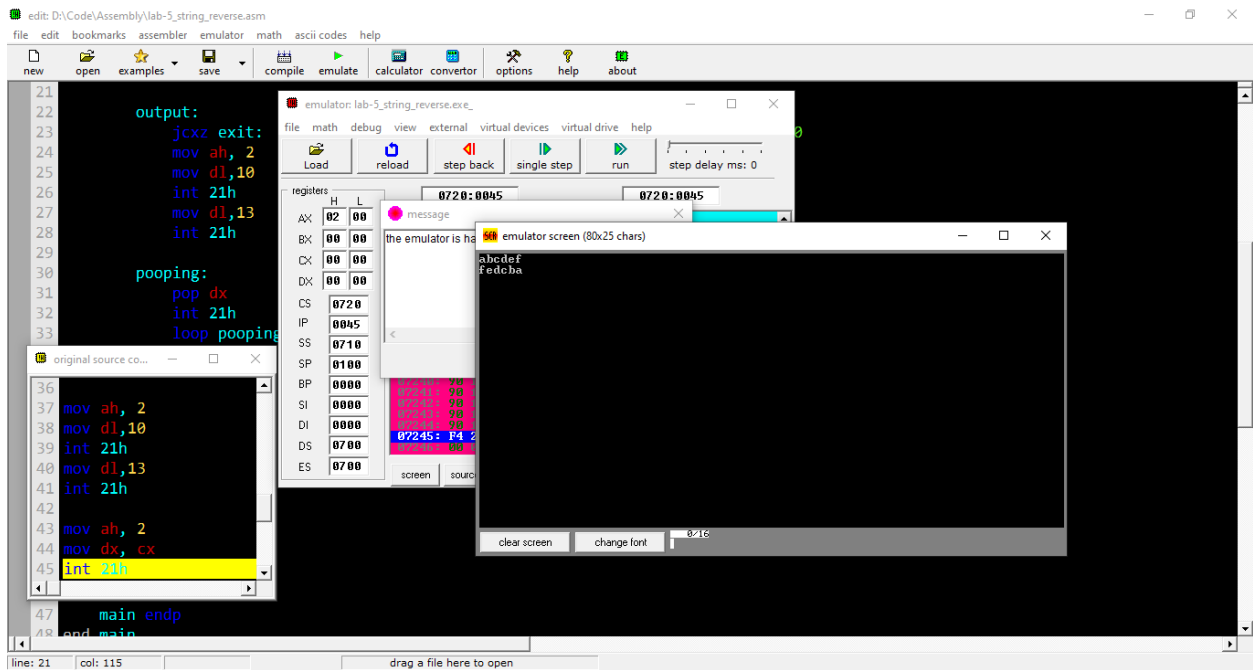
edit: D:\Code\Assembly\lab-5_string_reverse.asm

file edit bookmarks assembler emulator math ascii codes help

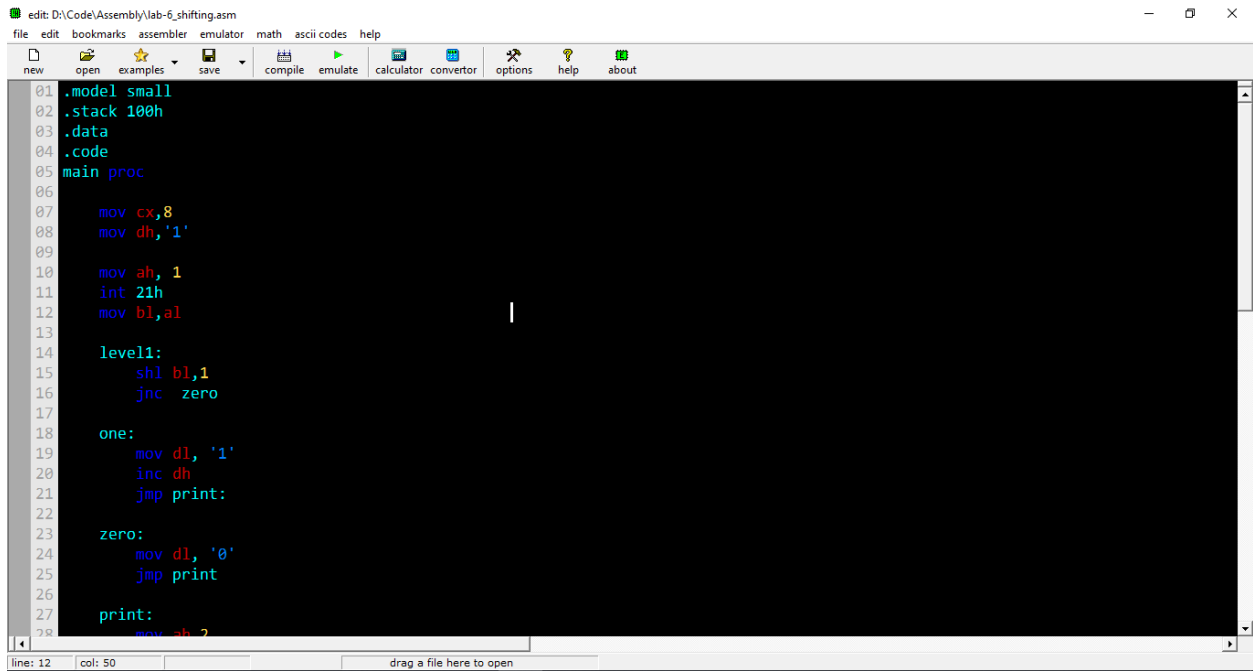
new open examples save compile emulate calculator convertor options help about

```
21 |
22 |    output:
23 |    jcxz exit:    ; JCXZ tests the contents of the CX register for 0
24 |    mov ah, 2
25 |    mov dl,10
26 |    int 21h
27 |    mov dl,13
28 |    int 21h
29 |
30 |    pooping:
31 |    pop dx
32 |    int 21h    ; connecting console
33 |    loop pooping
34 |
35 |    exit:
36 |
37 |    mov ah, 2
38 |    mov dl,10
39 |    int 21h
40 |    mov dl,13
41 |    int 21h
42 |
43 |    mov ah, 2
44 |    mov dx, cx
45 |    int 21h
46 |
47 |    main endp
48 |end main
```

line: 21 col: 115 drag a file here to open



Show shifting by assembly



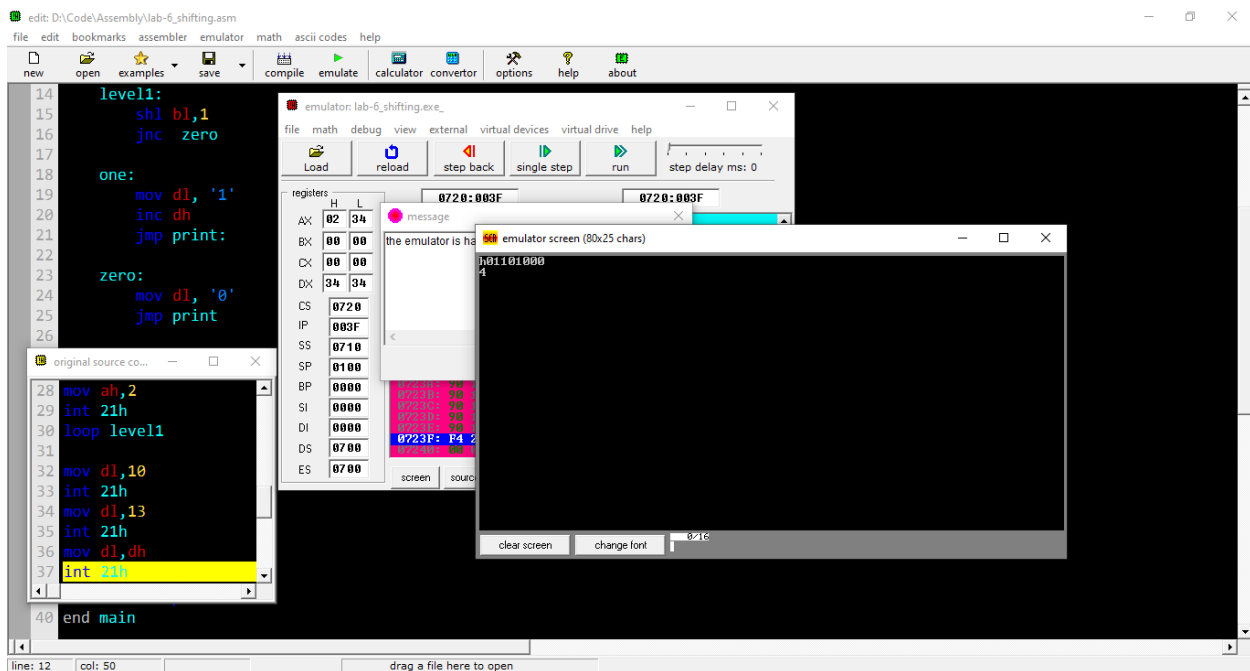
edit: D:\Code\Assembly\lab-6_shifting.asm

file edit bookmarks assembler emulator math ascii codes help

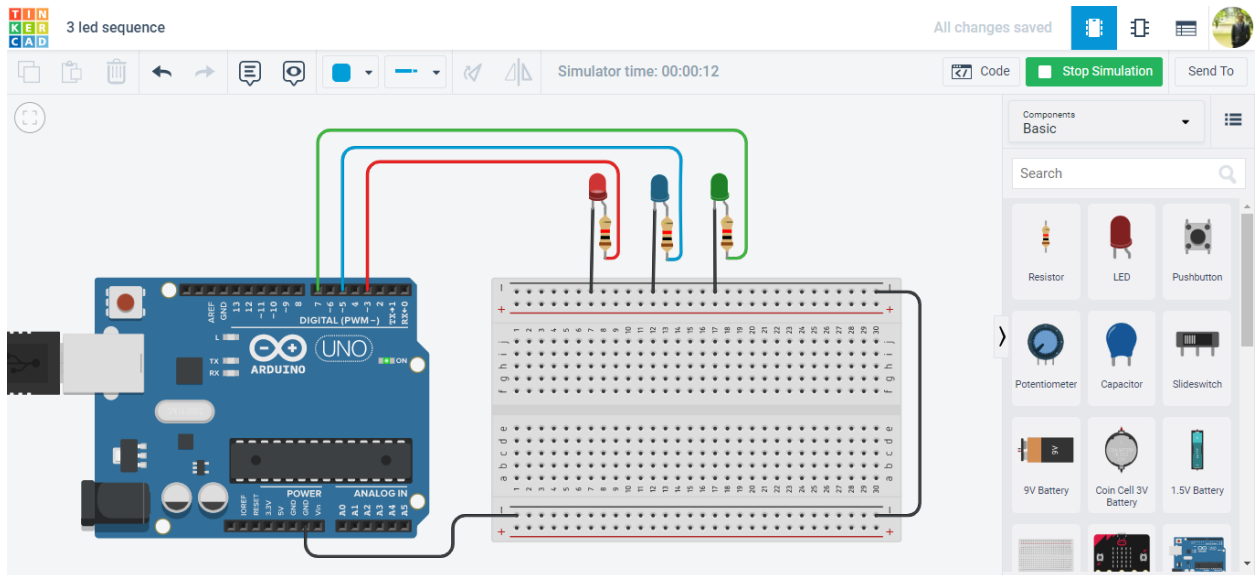
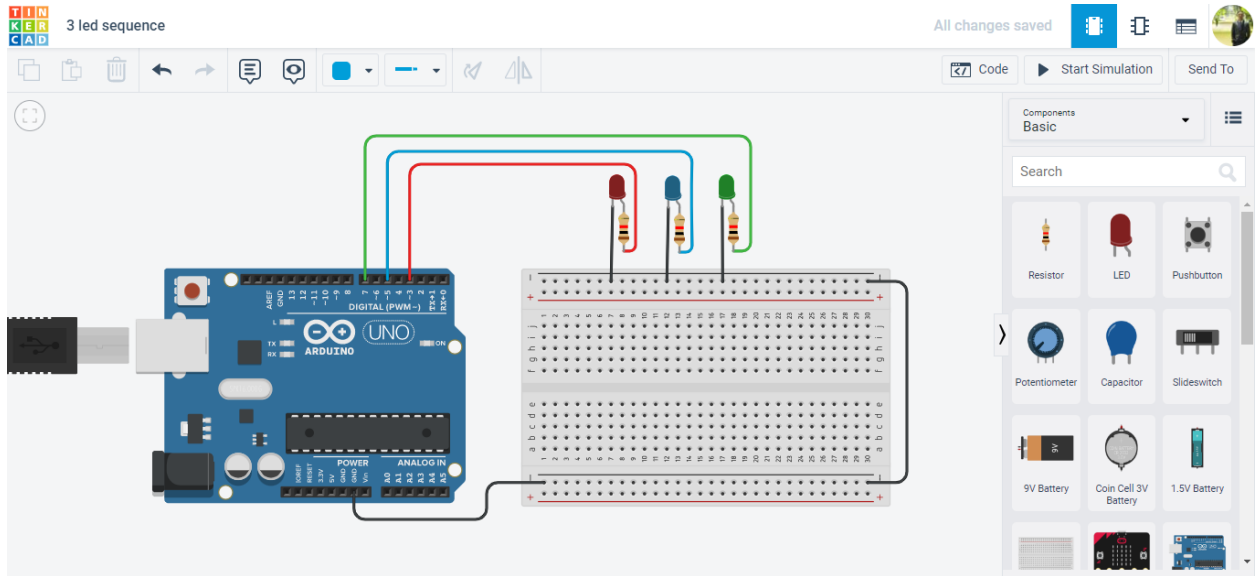
new open examples save compile emulate calculator convertor options help about

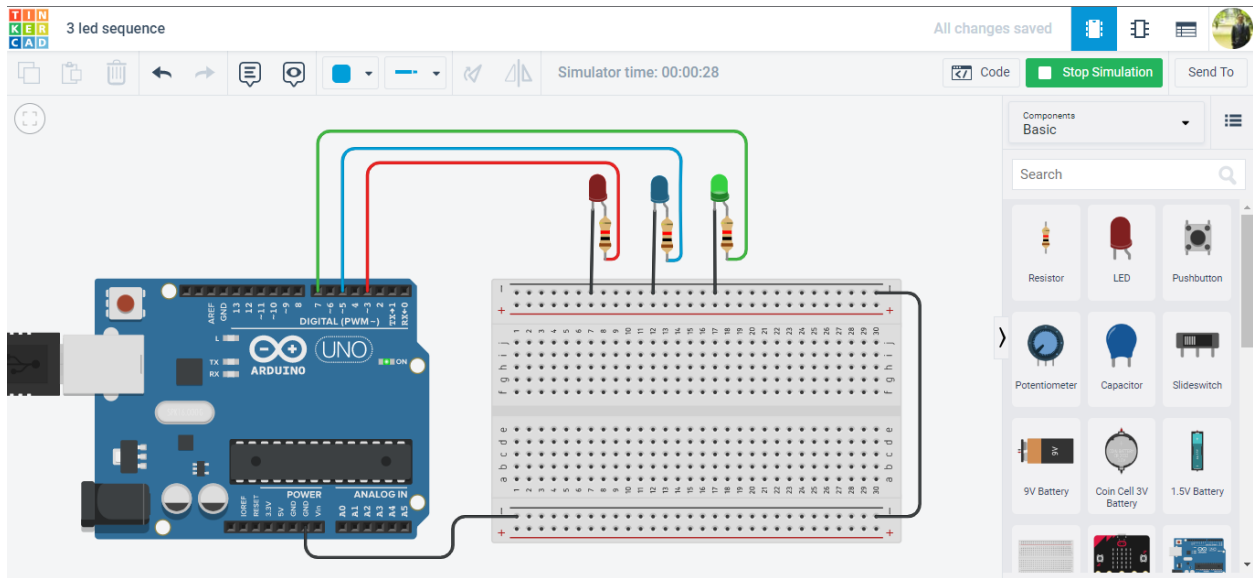
```
14 level1:
15     shl bl,1
16     jnc zero
17
18 one:
19     mov dl, '1'
20     inc dh
21     jmp print:
22
23 zero:
24     mov dl, '0'
25     jmp print
26
27 print:
28     mov ah,2
29     int 21h
30     loop level1
31
32     mov dl,10
33     int 21h
34     mov dl,13
35     int 21h
36     mov dl,dh
37     int 21h
38
39     main endp
40 end main
```

line: 12 col: 50 drag a file here to open

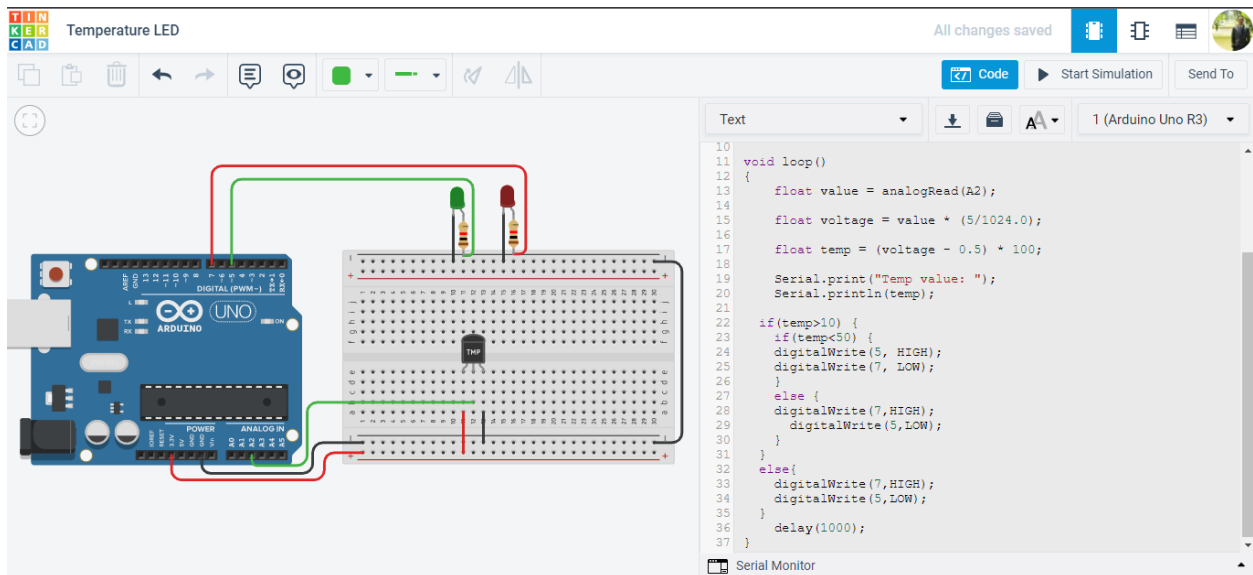


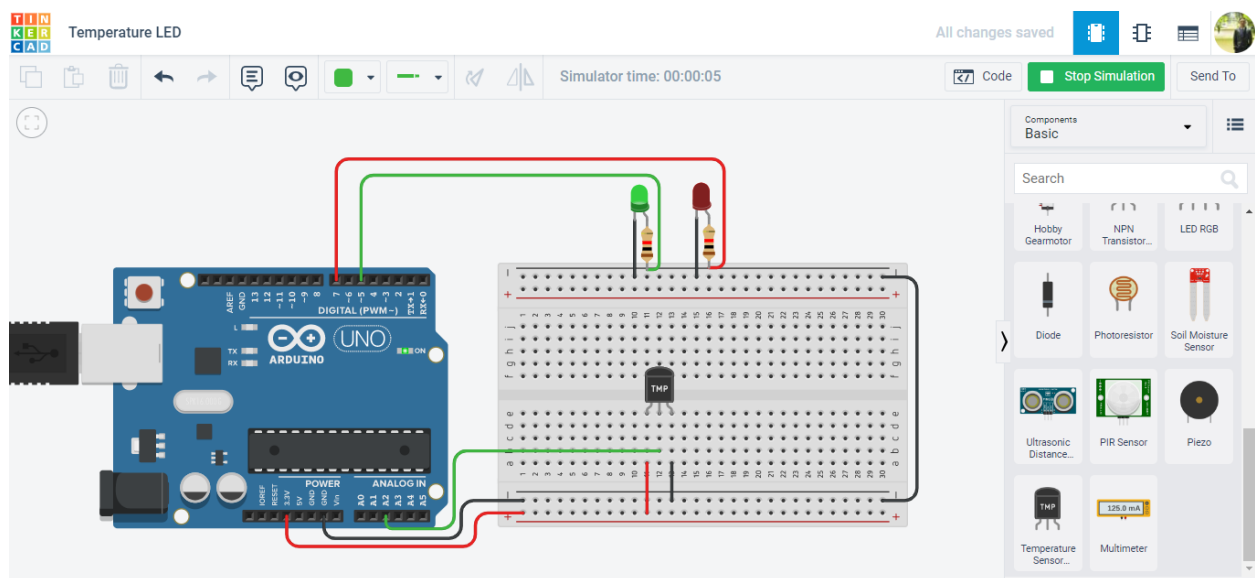
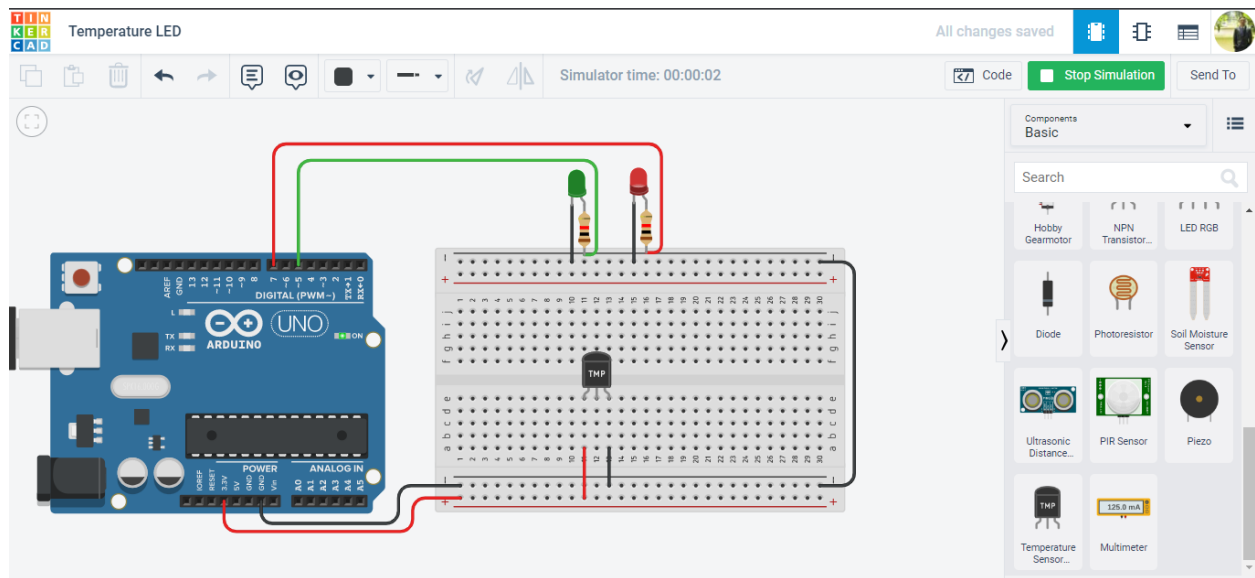
3 sequences of LED





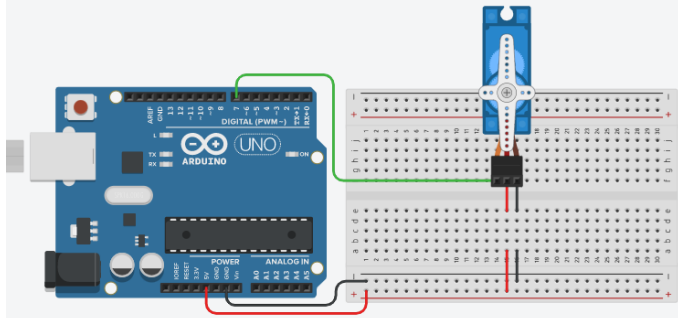
Turning LED on/off by TMP sensor





Rotation with servo motor

TIN K E R C A D Servo Motor All changes saved



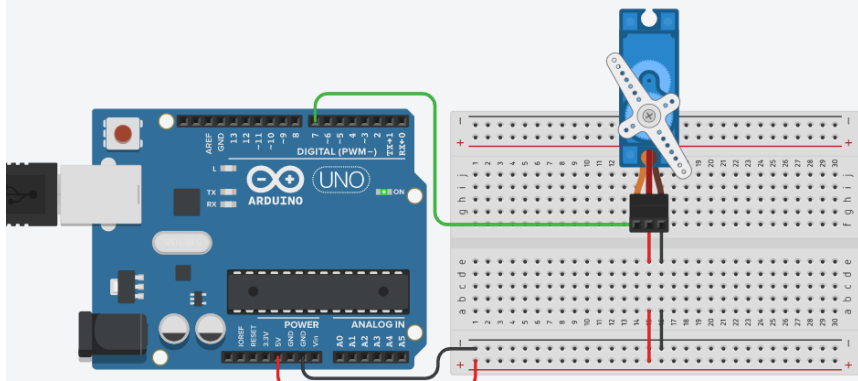
Code **Start Simulation** **Send To**

Text 1 (Arduino Uno R3)

```
1 #include<Servo.h>
2
3 Servo shanto;
4
5 void setup()
6 {
7   shanto.attach(7);
8 }
9
10 void loop()
11 {
12   shanto.write(0);
13   delay(800);
14   shanto.write(10);
15   delay(800);
16   shanto.write(20);
17   delay(800);
18   shanto.write(30);
19   delay(800);
20   shanto.write(40);
21   delay(800);
22   shanto.write(50);
23   delay(800);
24   shanto.write(60);
25   delay(800);
26   shanto.write(70);
27   delay(800);
28   shanto.write(80);
29 }
```

Serial Monitor

TIN K E R C A D Servo Motor All changes saved

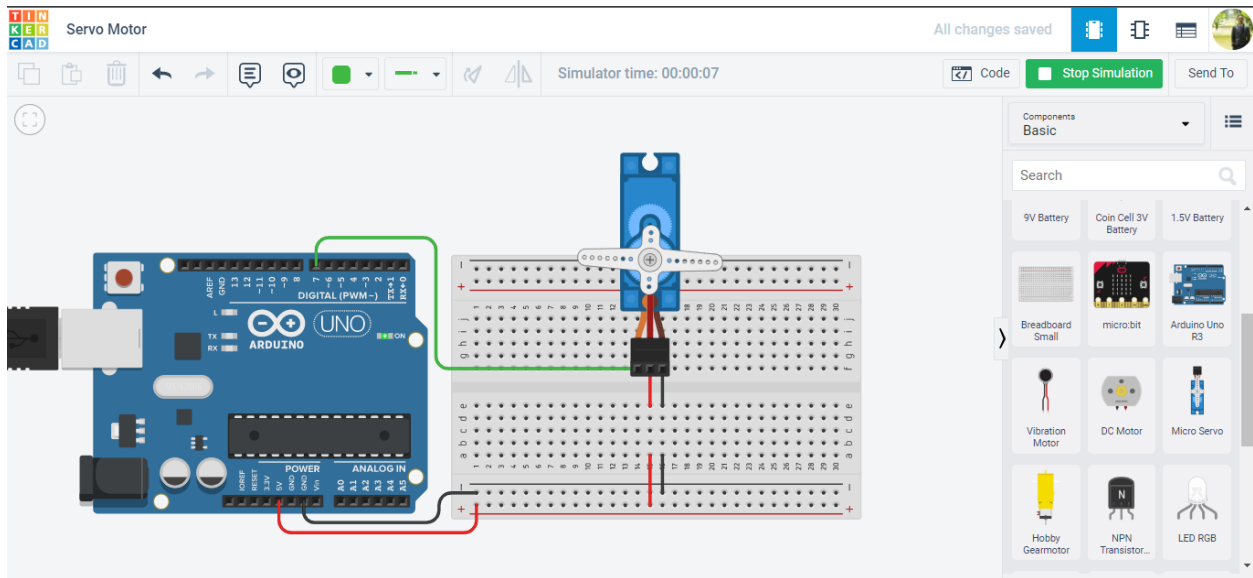


Code **Stop Simulation** **Send To** Simulator time: 00:00:03

Components Basic

Search

- 9V Battery
- Coin Cell 3V Battery
- 1.5V Battery
- Breadboard Small
- micro:bit
- Arduino Uno R3
- Vibration Motor
- DC Motor
- Micro Servo
- Hobby Gearmotor
- NPN Transistor...
- LED RGB



Detection by ultrasonic distance sensor

TINKER CAD Frantic Juttuli

All changes saved

Code Start Simulation Send To

Text

1 (Arduino Uno R3)

```

1
2 int trig = 7;
3 int echo = 4;
4 long duration;
5 int distance;
6
7 void setup() {
8   pinMode(trig, OUTPUT);
9   pinMode(echo, INPUT);
10
11   Serial.begin(9600);
12
13 }
14 void loop() {
15   digitalWrite(trig, LOW);
16   delay(1);
17   digitalWrite(trig, HIGH);
18   delay(1);
19   digitalWrite(trig, LOW);
20
21   duration = pulseIn(echo, HIGH);
22
23   distance = (0.034 * duration) / 2;
24
25   Serial.print("Distance in CM: ");
26   Serial.println(distance);
27
28   delay(1000);

```

Serial Monitor

