Microprocessor, Embedded Systems and IoT

LAB REPORT

Course Code: CSE 232

Submitted to:

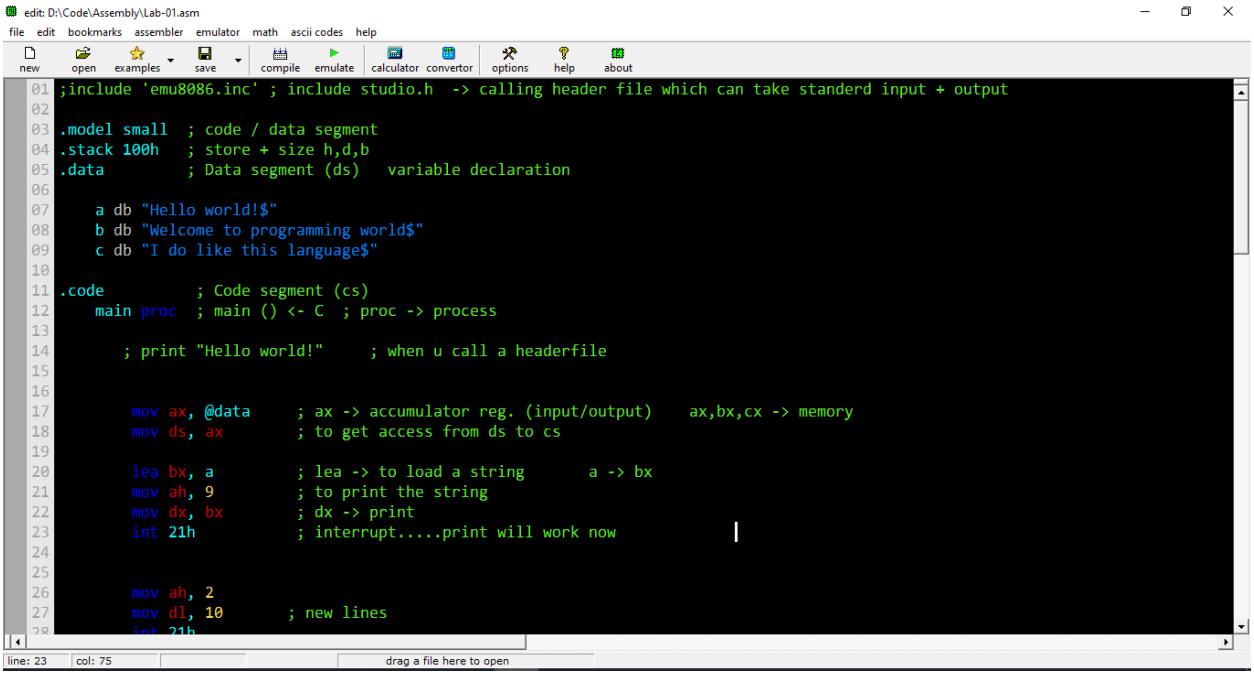
Sonia Nasrin
Lecturer
Daffodil International University

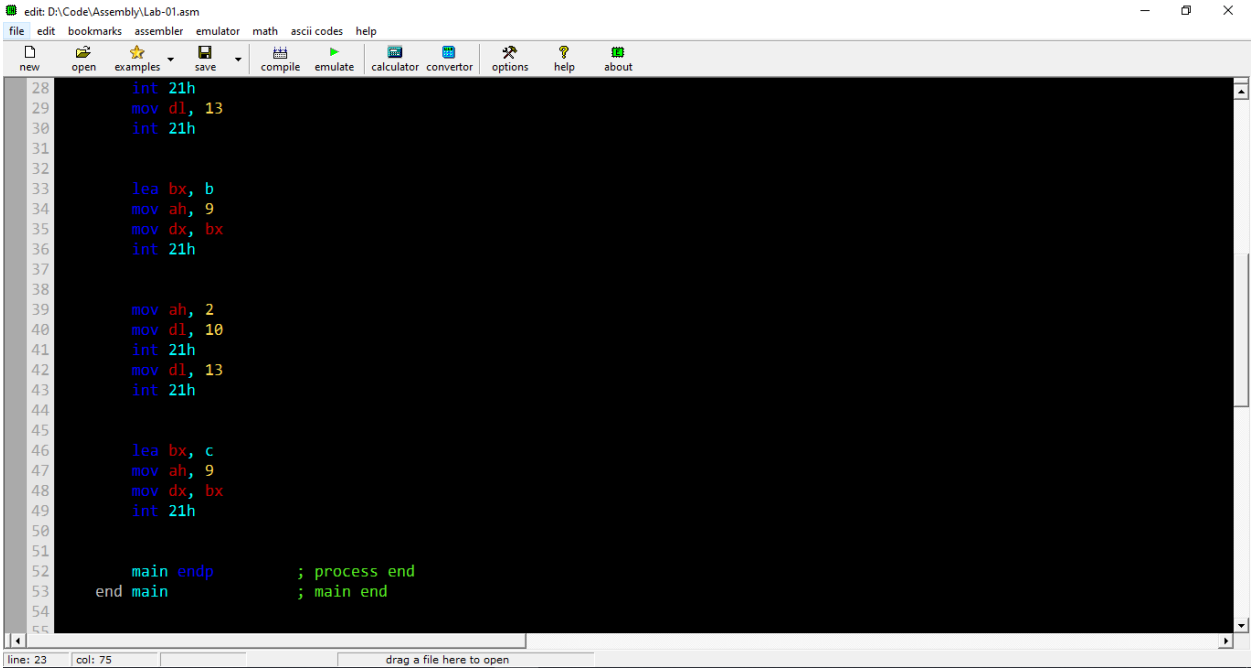
Date of Submission: 12/8/2022

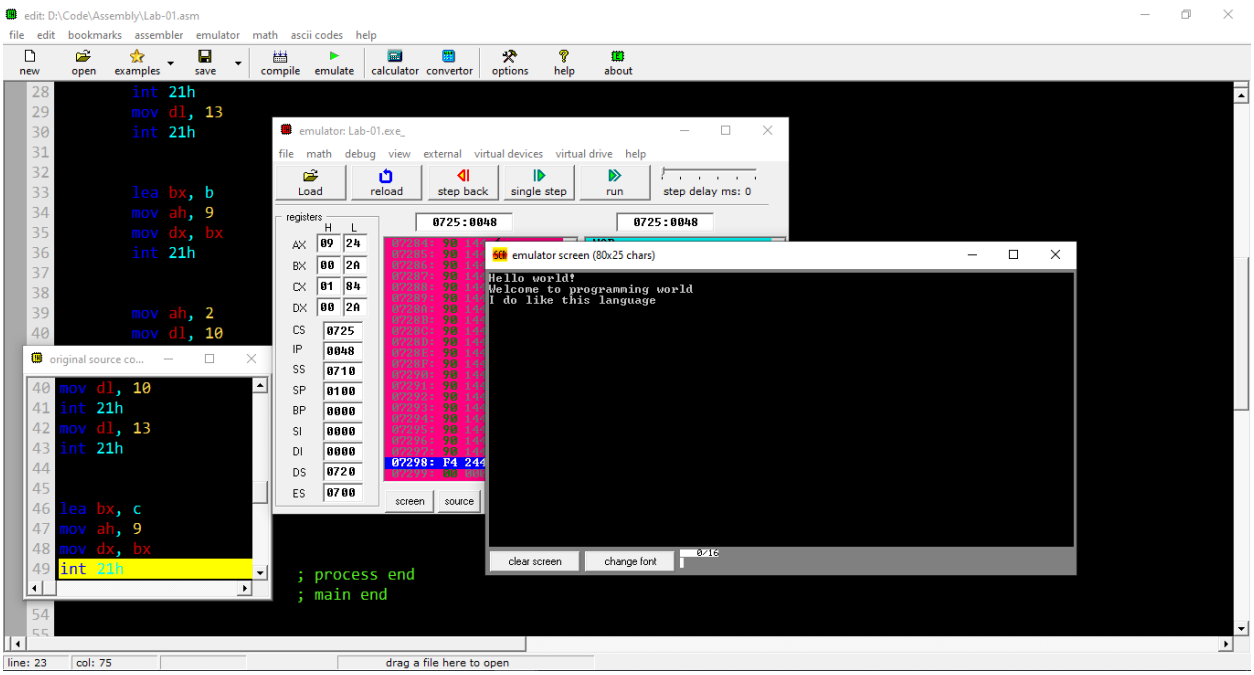
Submitted by:

MD. RAKIBUL ISLAM SHANTO
ID no: 203-15-3871
Section: PC-A
Department of CSE
Daffodil International University

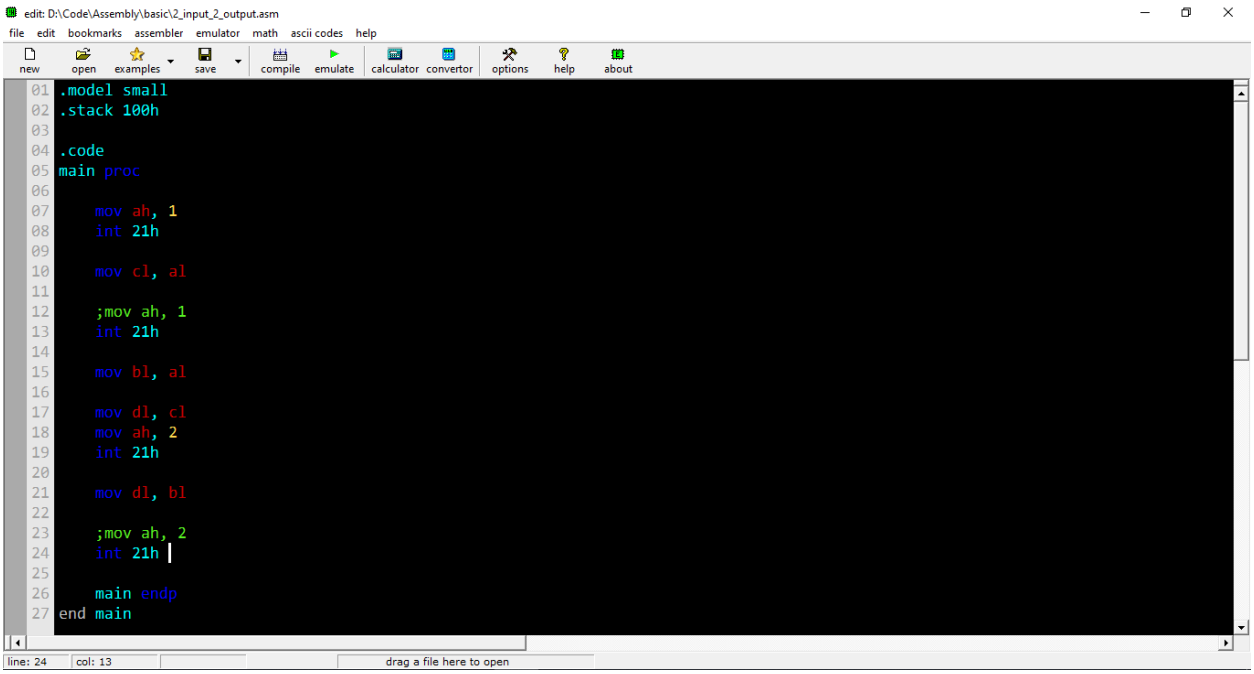
Printing 3 strings from data segment

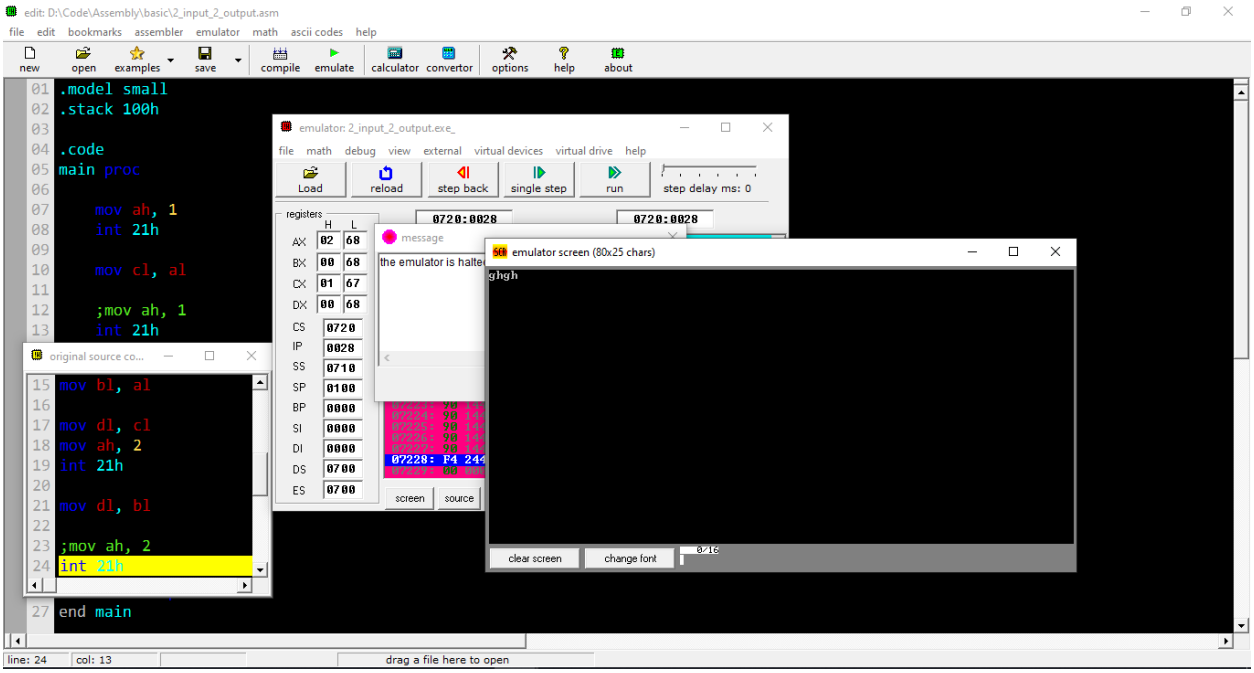




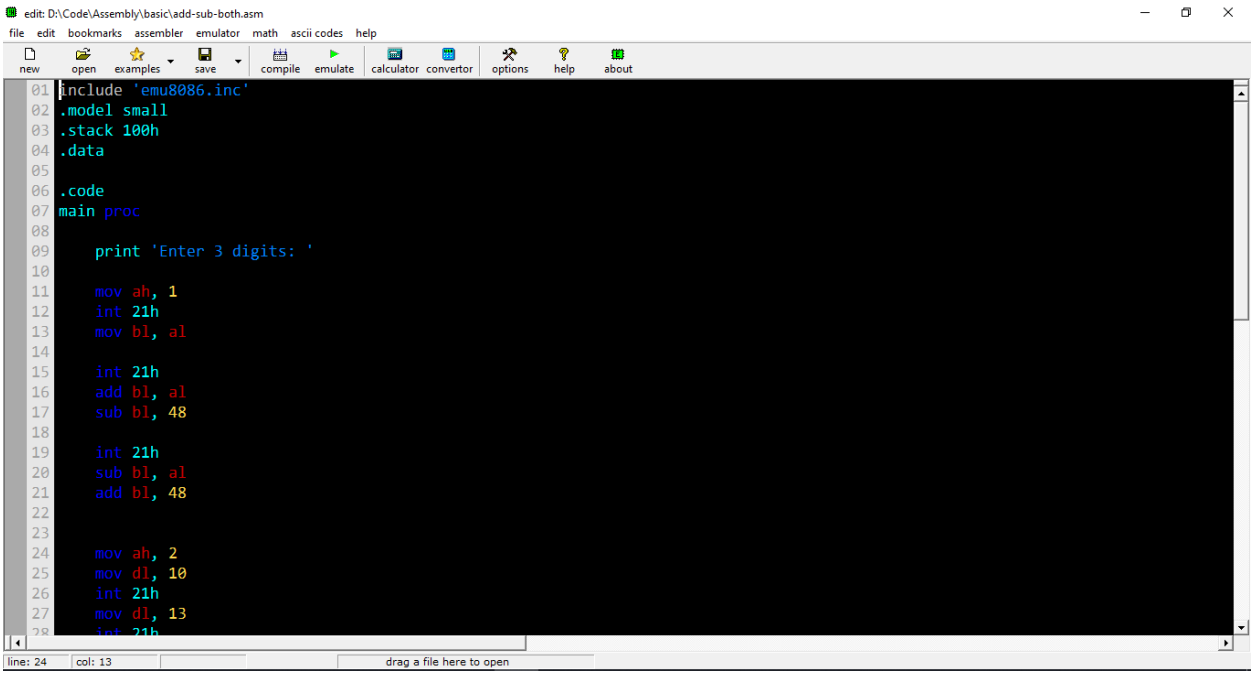


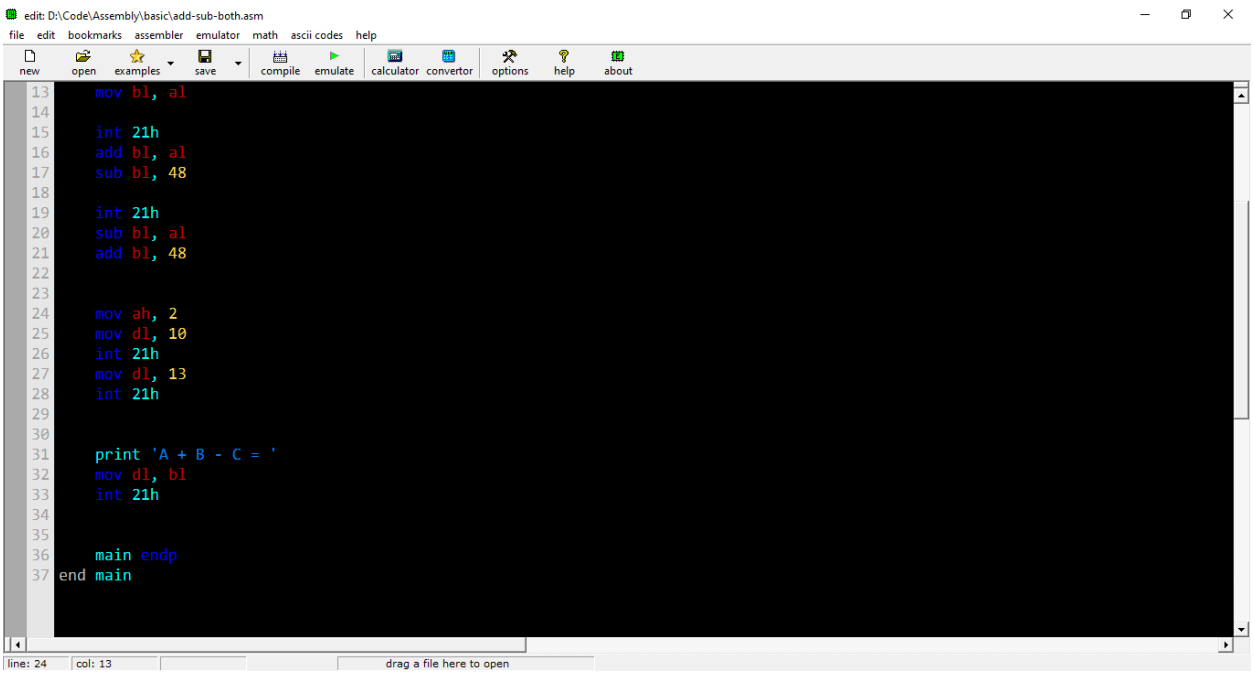
Taking input and printing output

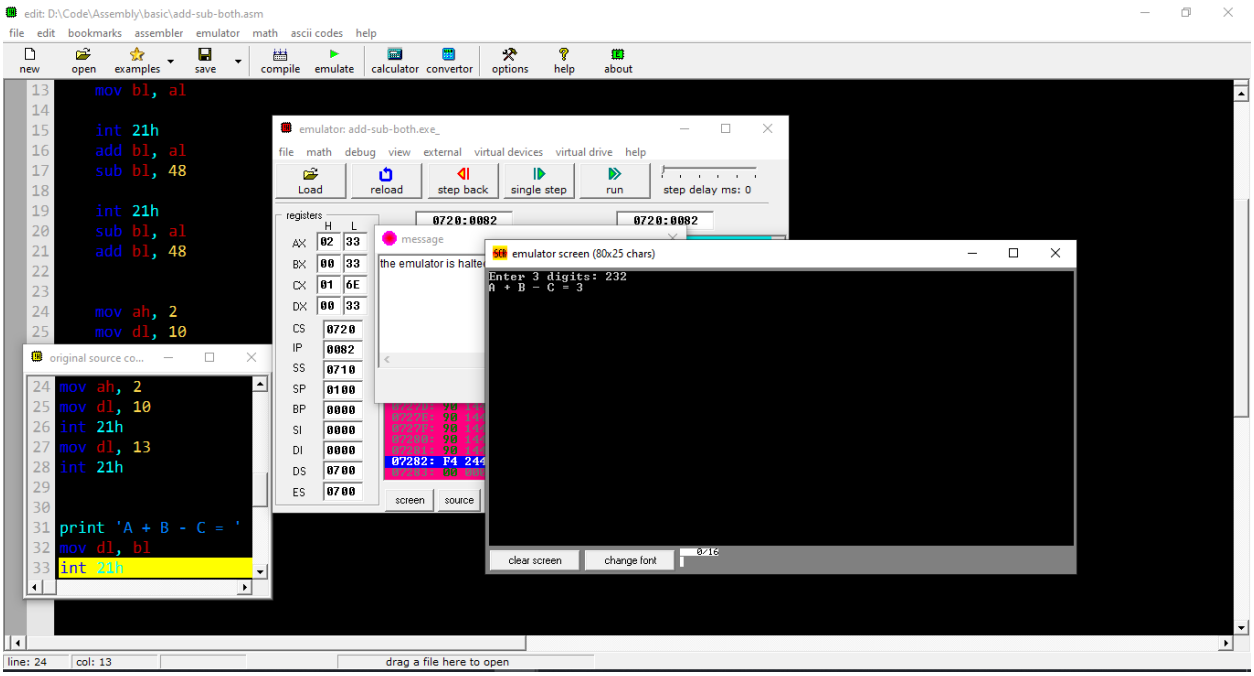




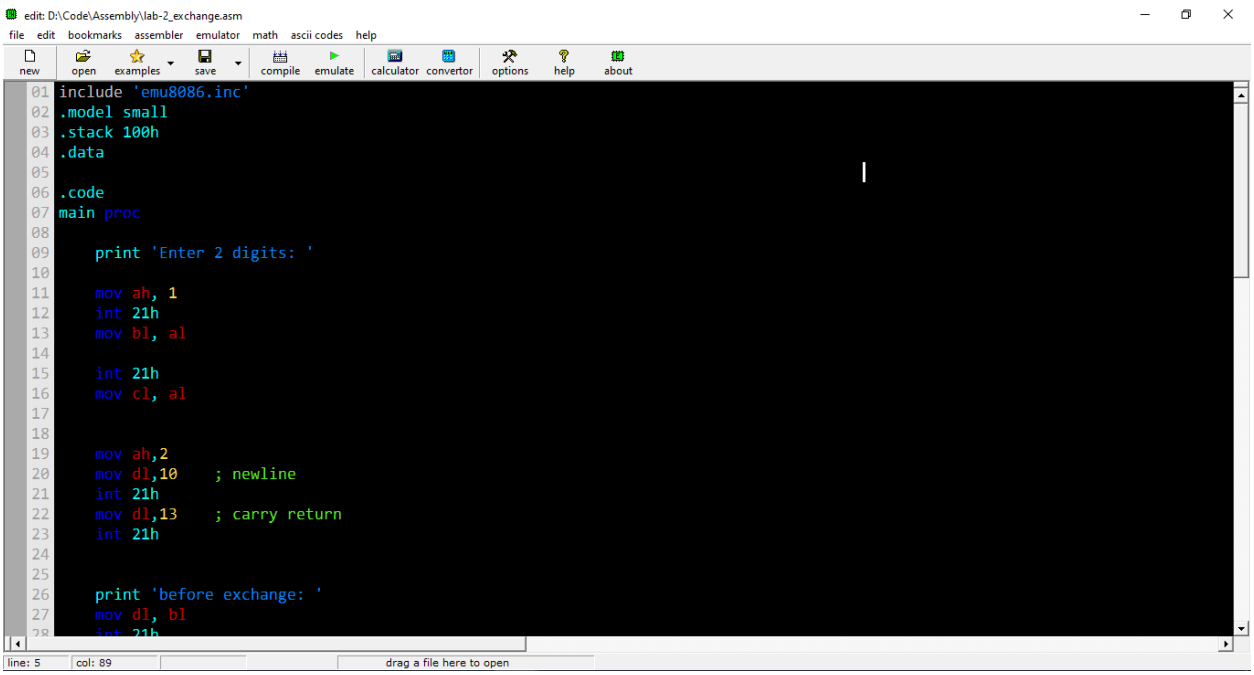
Addition & Subtraction

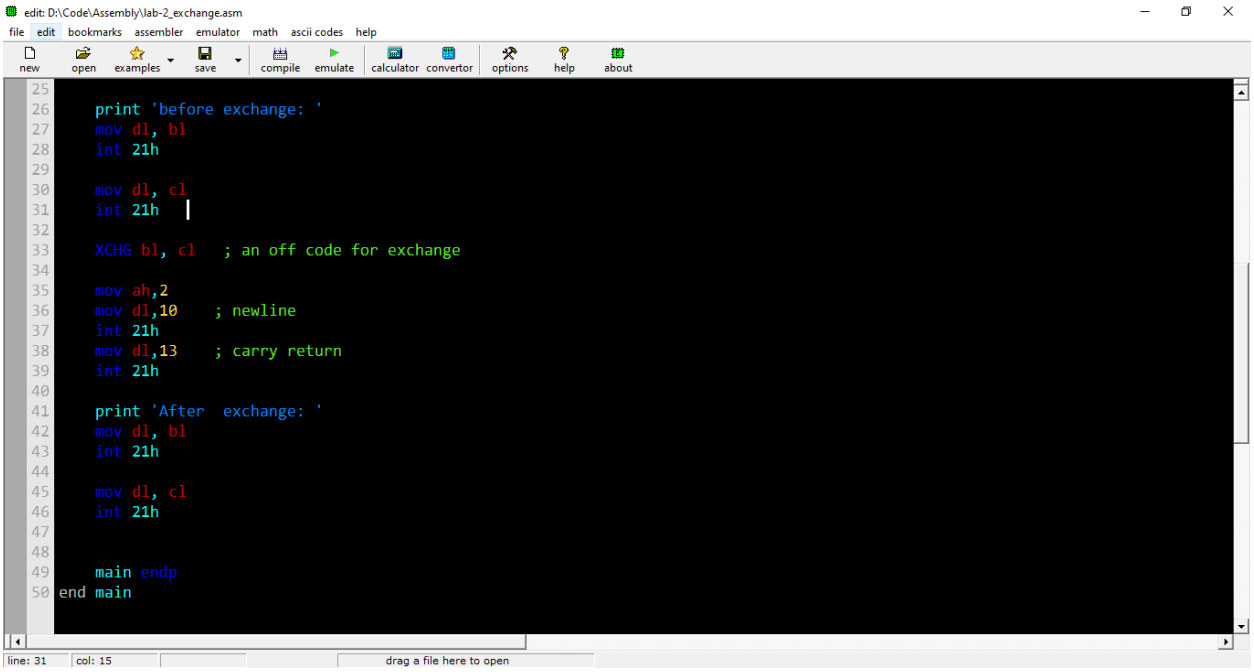


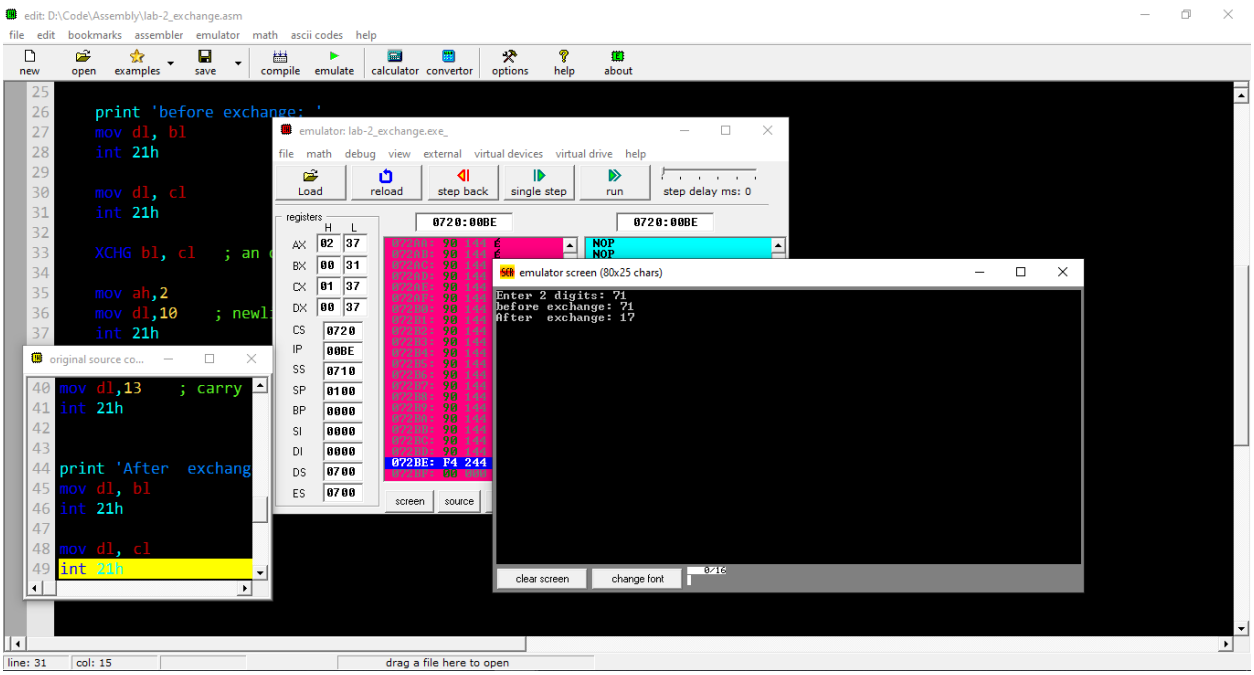




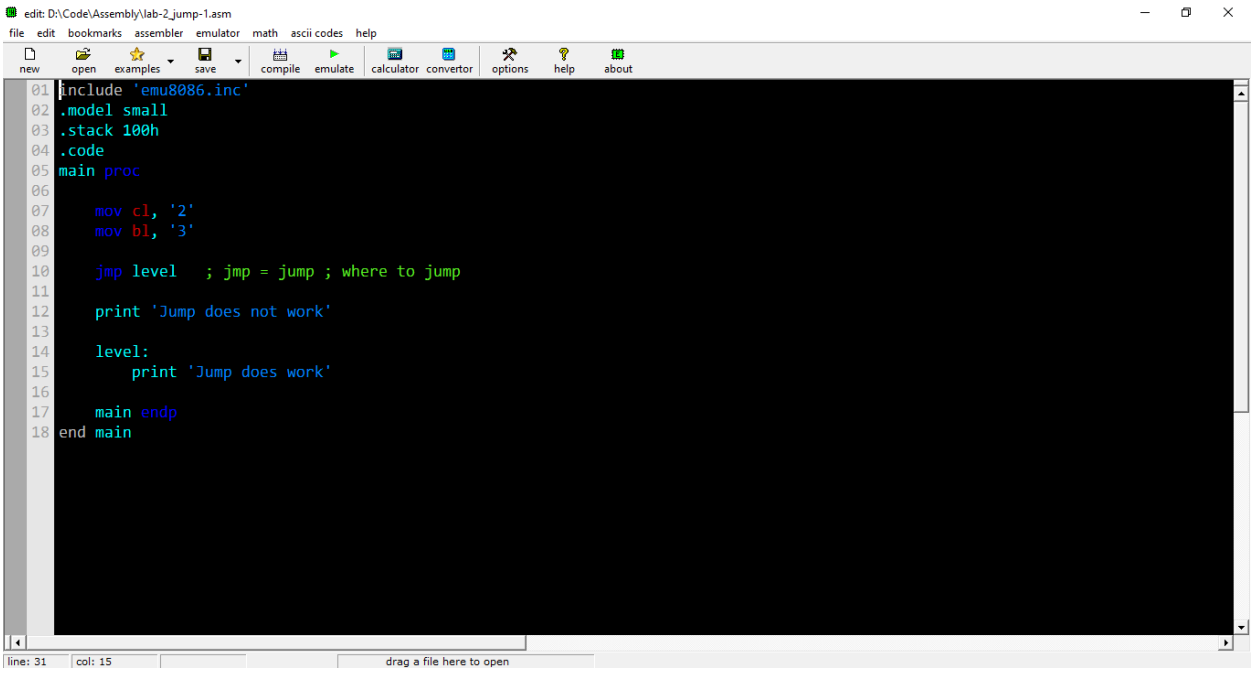
Exchange between 2 numbers

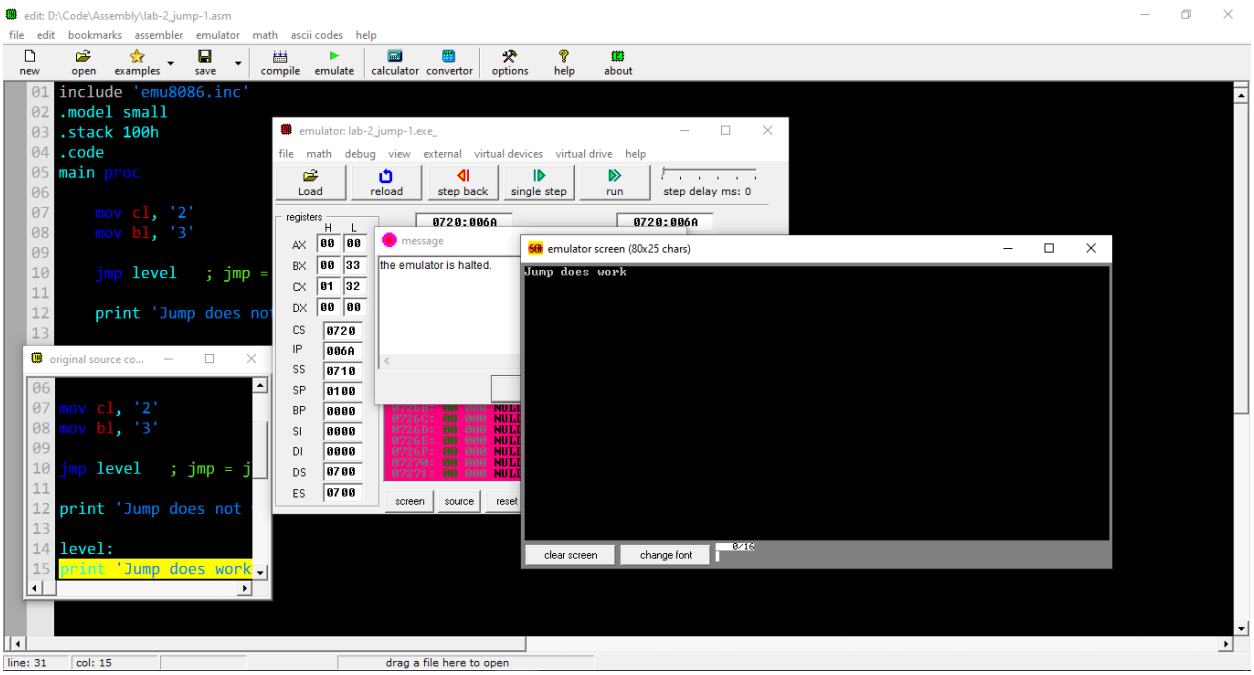

edit: D:\Code\Assembly\lab-2_exchange.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
01 include 'emu8086.inc'
02 .model small
03 .stack 100h
04 .data
05
06 .code
07 main proc
08
09 print 'Enter 2 digits: '
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
line: 5
col: 89
drag a file here to open


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
line: 31
col: 15
drag a file here to open

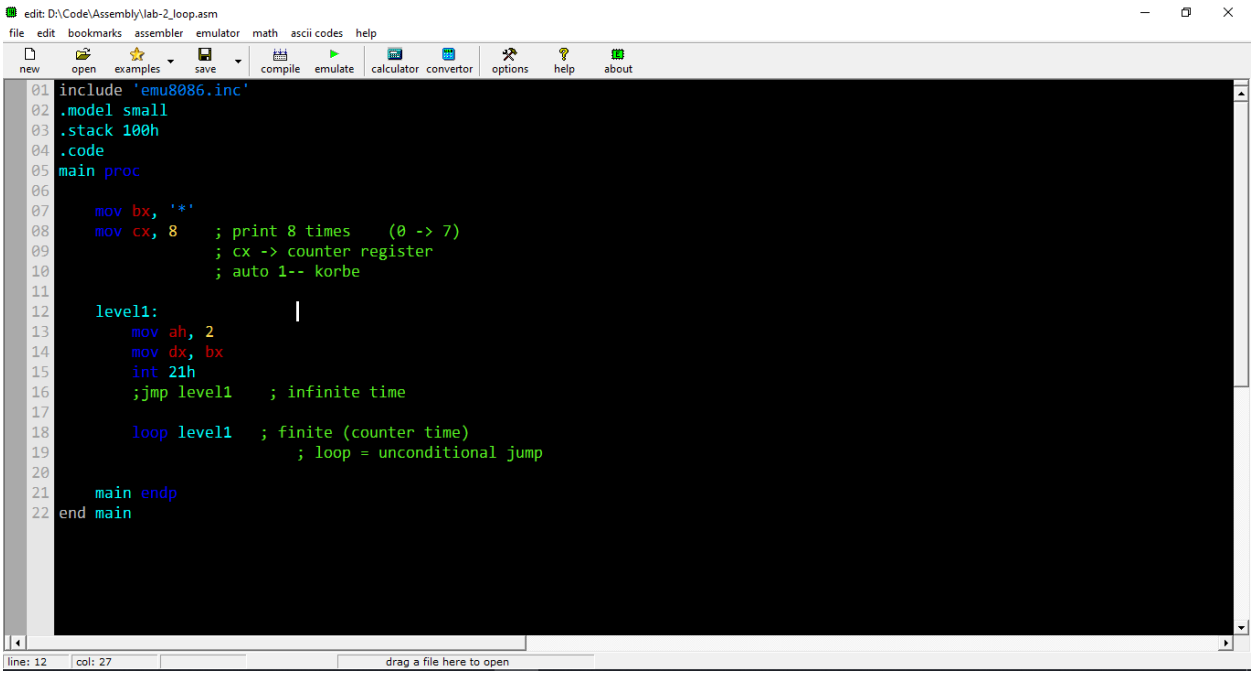

edit: D:\Code\Assembly\lab-2_exchange.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
print 'before exchange: '
mov dl, bl
int 21h
mov dl, cl
int 21h
XCHG bl, cl ; an
mov ah,2
mov dl,10 ; newl
int 21h
emulator: lab-2_exchange.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 37
BX 00 31
CX 01 37
DX 00 37
CS 0720
IP 00BE
SS 0710
SP 0100
BP 0000
SI 0000
DI 0000
DS 0700
ES 0700
0720:00BE
0720:00BE
NOP
NOP
072BE: F4 244
screen source
original source co...
mov dl,13 ; carry
int 21h
print 'After exchang
mov dl, bl
int 21h
mov dl, cl
int 21h
emulator screen (80x25 chars)
Enter 2 digits: 71
before exchange: 71
After exchange: 17
clear screen change font
line: 31 col: 15
drag a file here to open

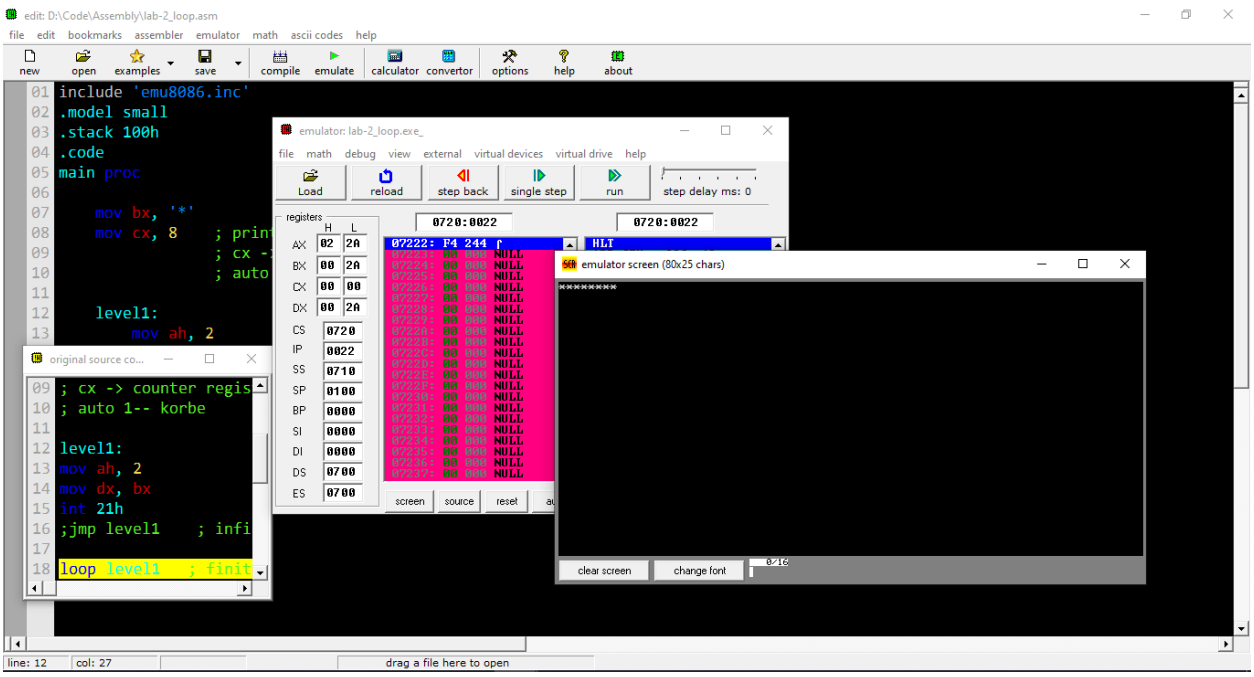
Use of ‘jmp’


edit: D:\Code\Assembly\lab-2_jump-1.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
include 'emu8086.inc'
.model small
.stack 100h
.code
main proc
mov cl, '2'
mov bl, '3'
jmp level ; jmp = jump ; where to jump
print 'Jump does not work'
level:
print 'Jump does work'
main endp
end main
line: 31 col: 15
drag a file here to open

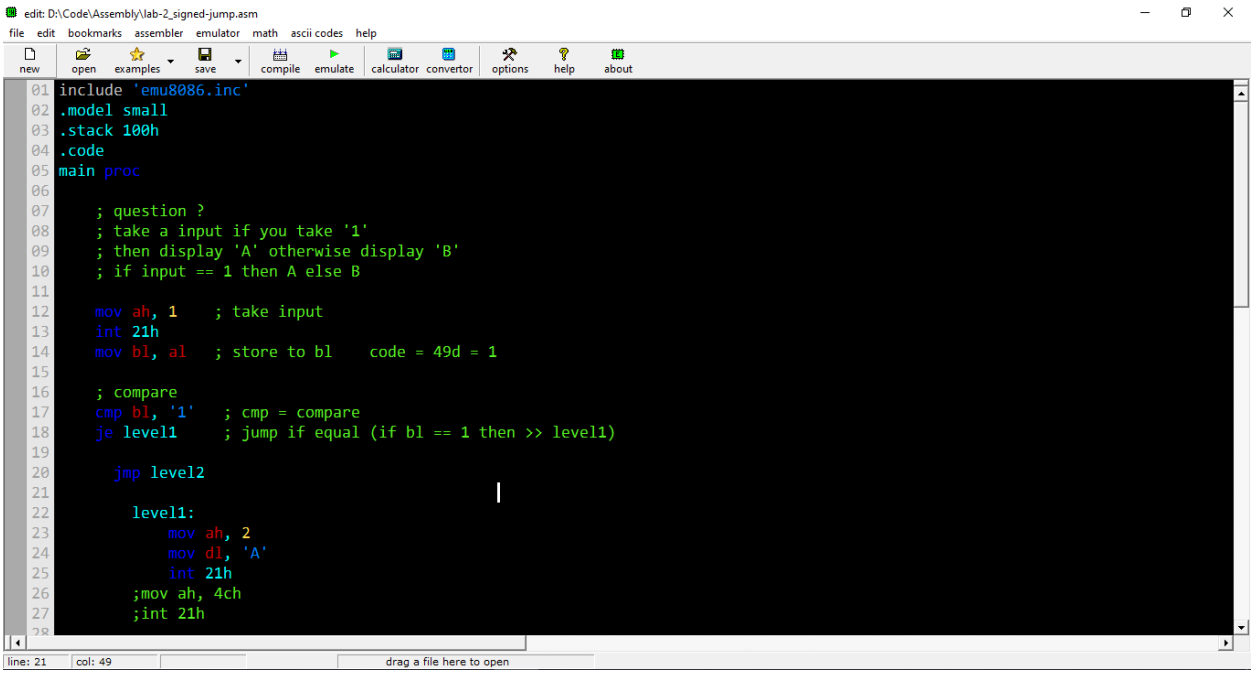


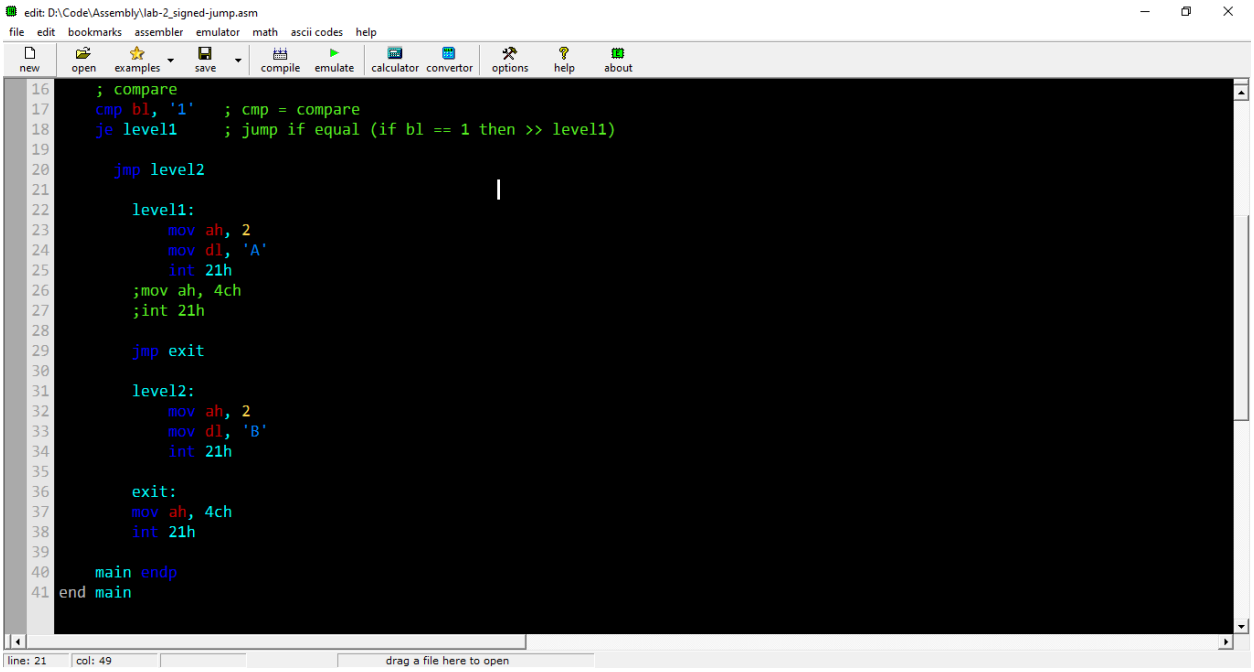
Print star(*) 8 times by for loop

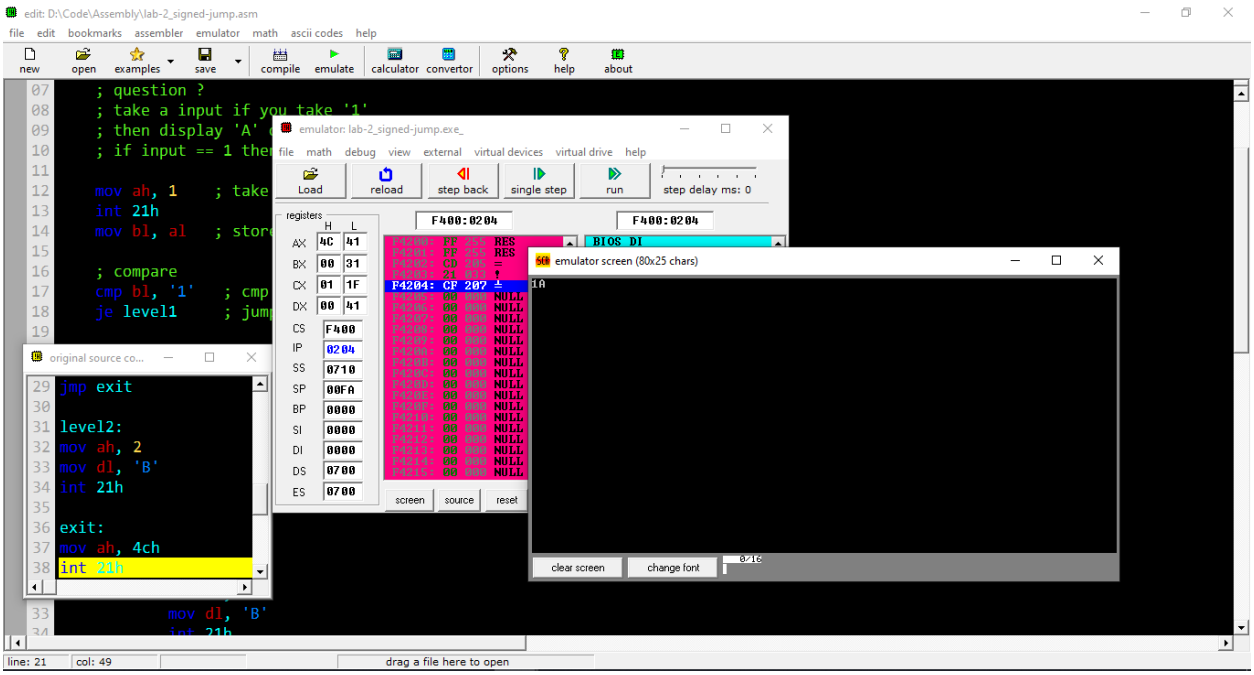




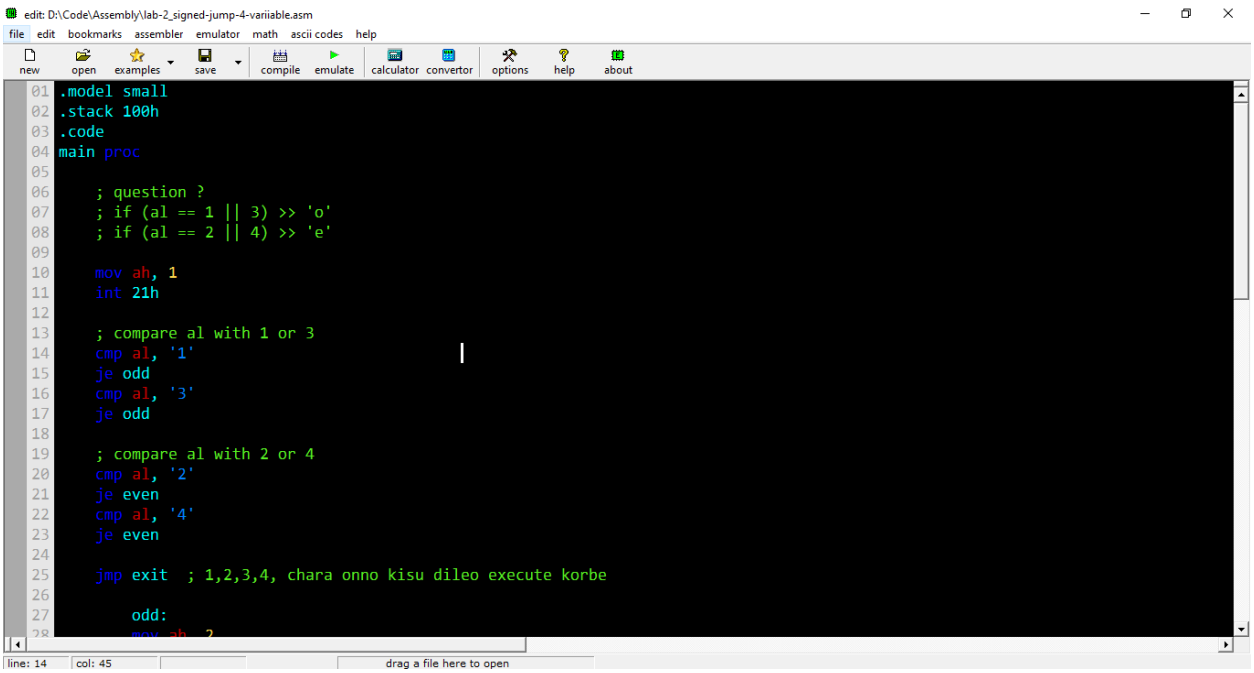
Take an input. If you take '1' then display 'A' otherwise display 'B'

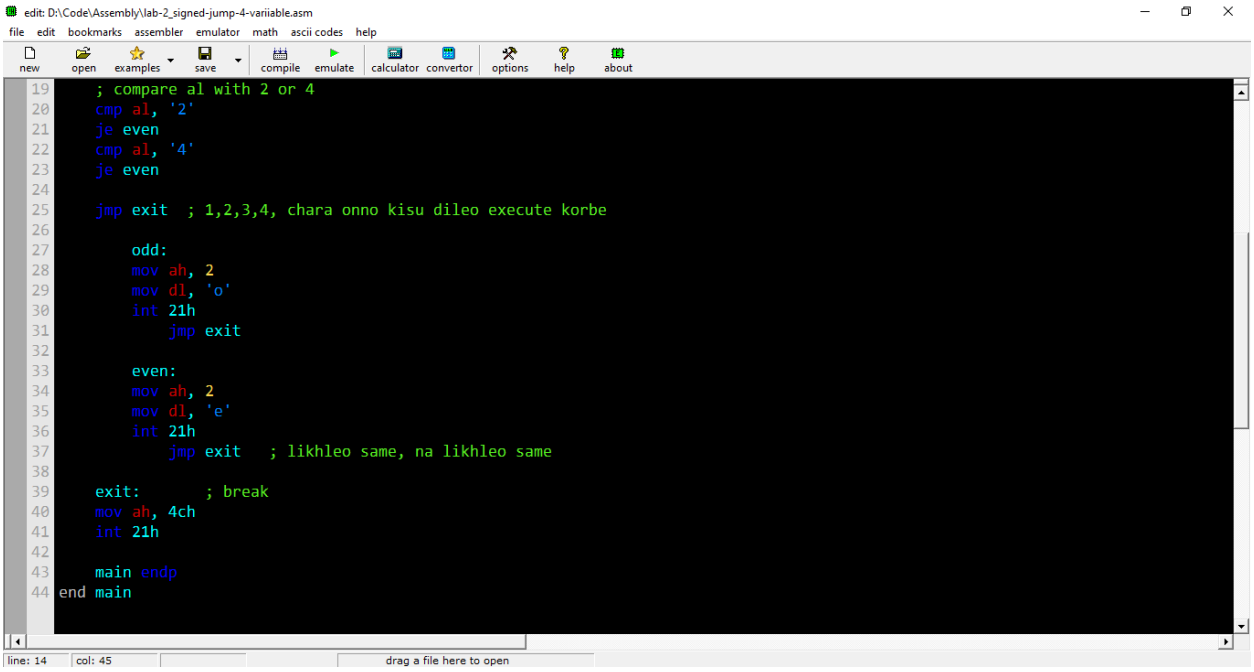


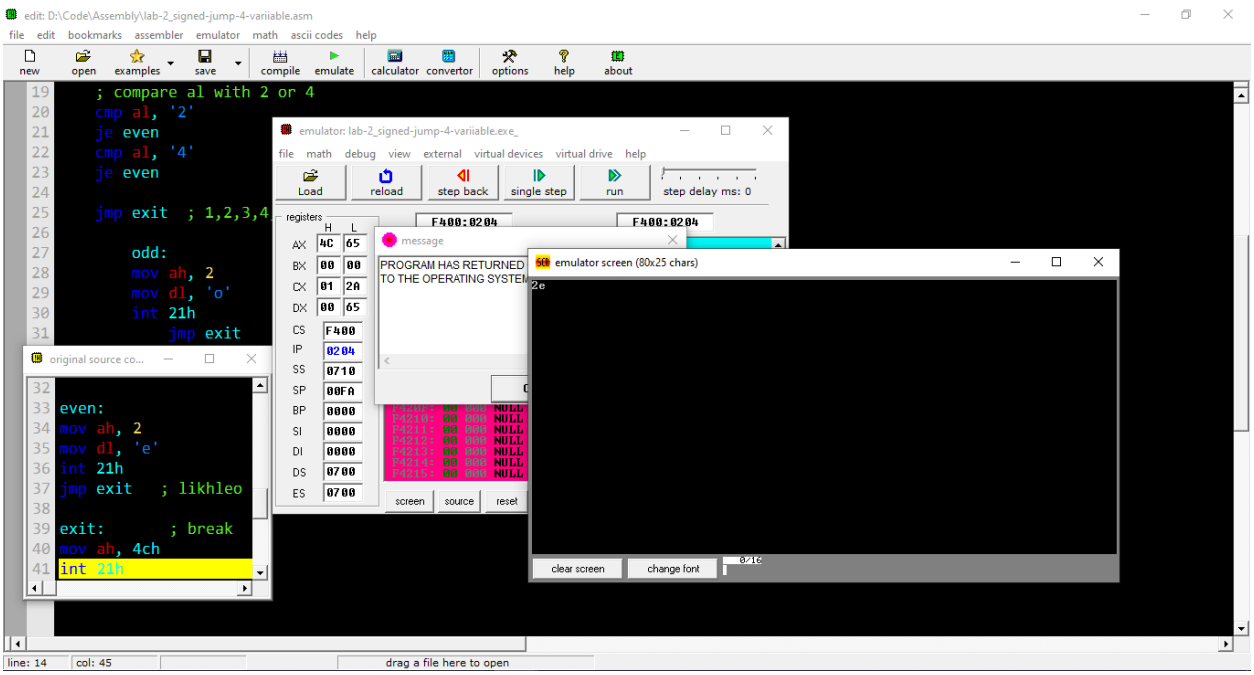




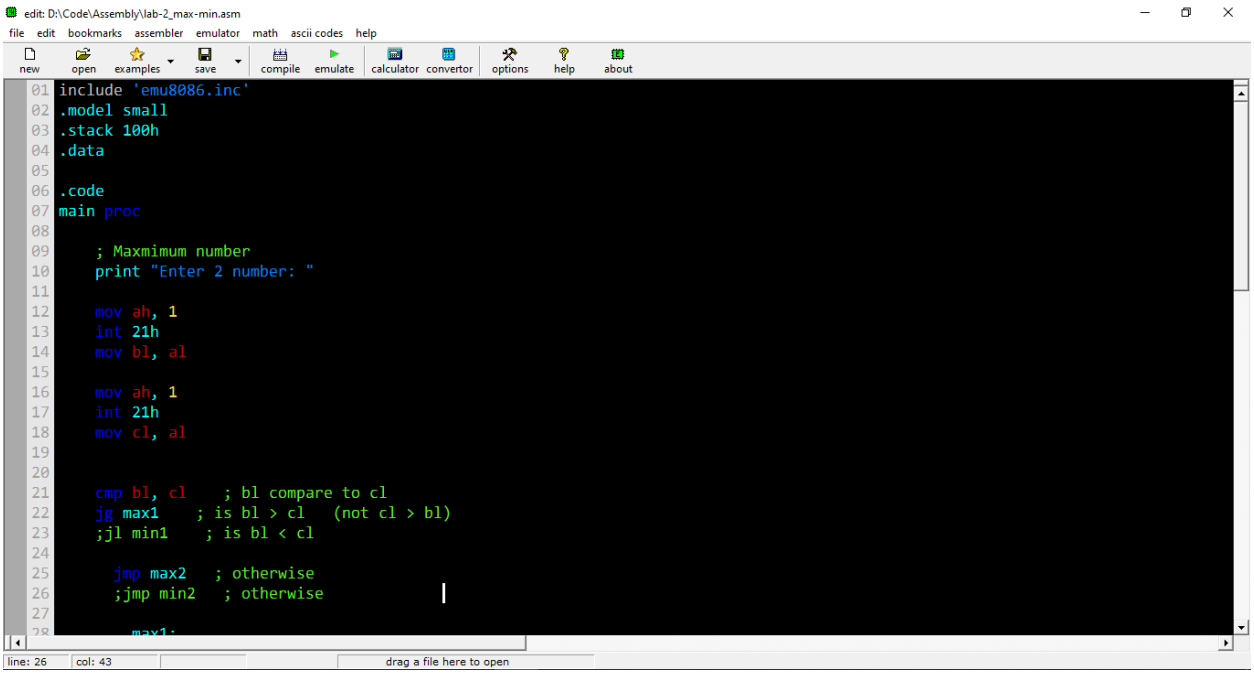
If input is 1 or 3 then print ‘o’ or if the input is 2 or 4 then print ‘e’

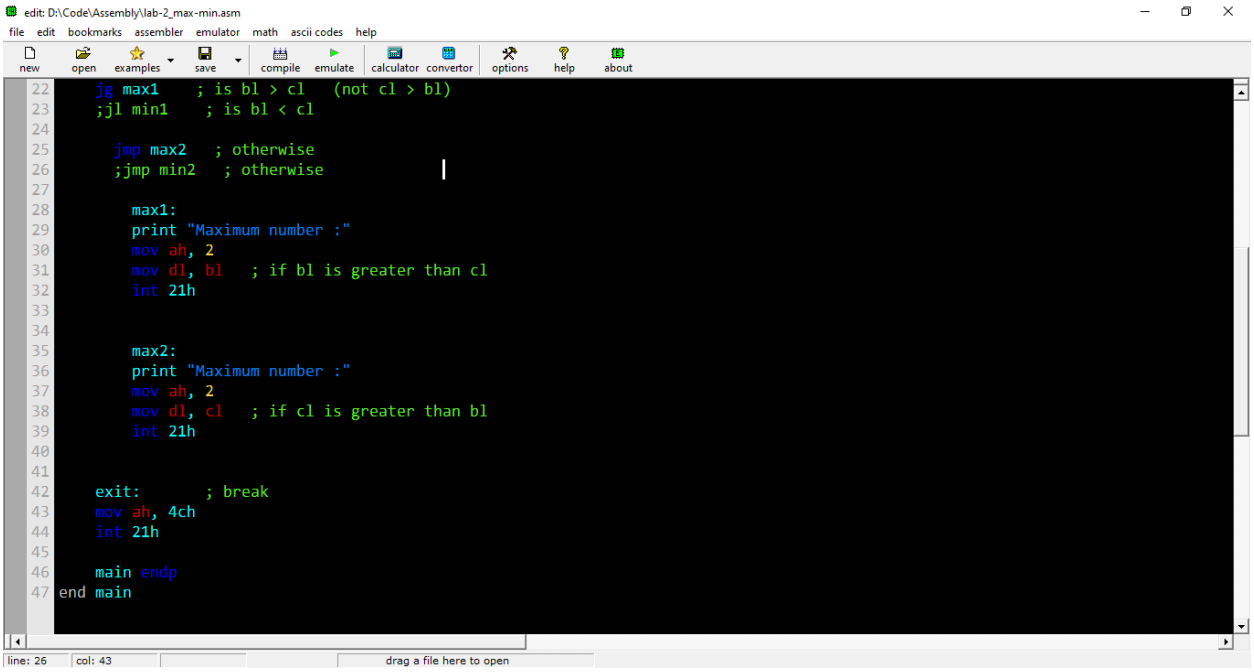

edit: D:\Code\Assembly\lab-2_signed-jump-4-variiable.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
01 .model small
02 .stack 100h
03 .code
04 main proc
05
06 ; question ?
07 ; if (al == 1 || 3) >> 'o'
08 ; 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 korbe
26
27 odd:
line: 14 col: 45
drag a file here to open

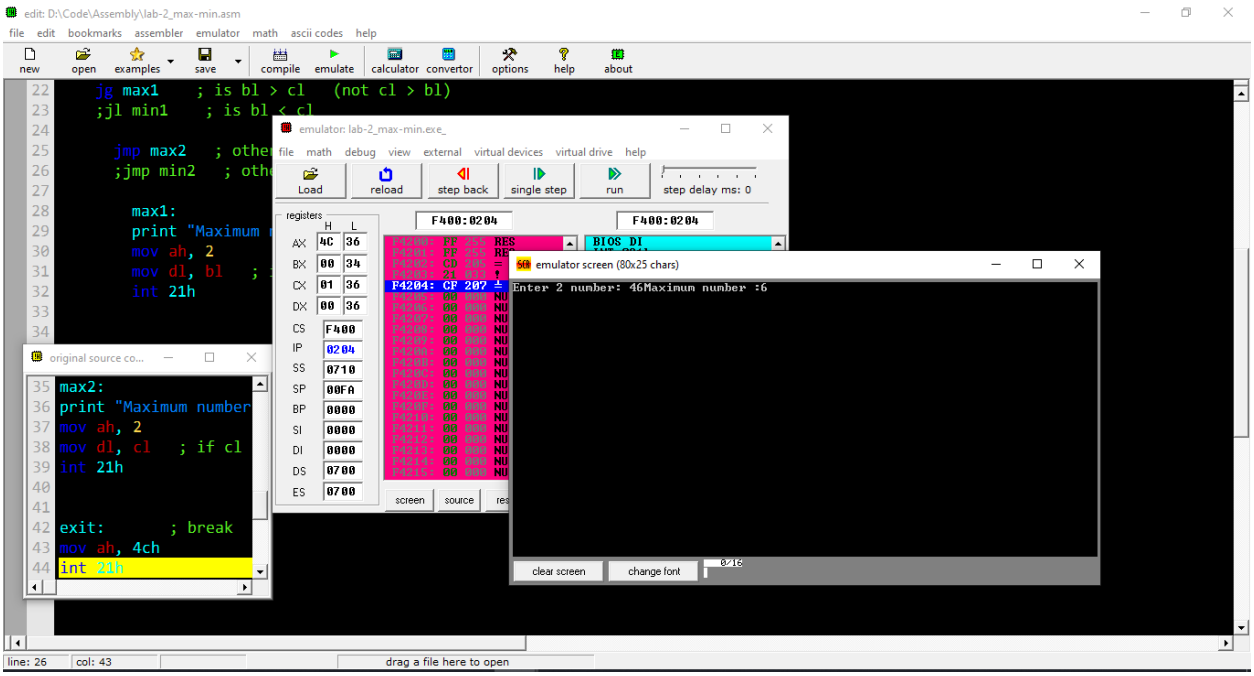

edit: D:\Code\Assembly\lab-2_signed-jump-4-variiable.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
19 ; compare al with 2 or 4
20 cmp al, '2'
21 je even
22 cmp al, '4'
23 je even
24
25 jmp exit ; 1,2,3,4, chara onno kisu dileo execute korbe
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
line: 14 col: 45
drag a file here to open



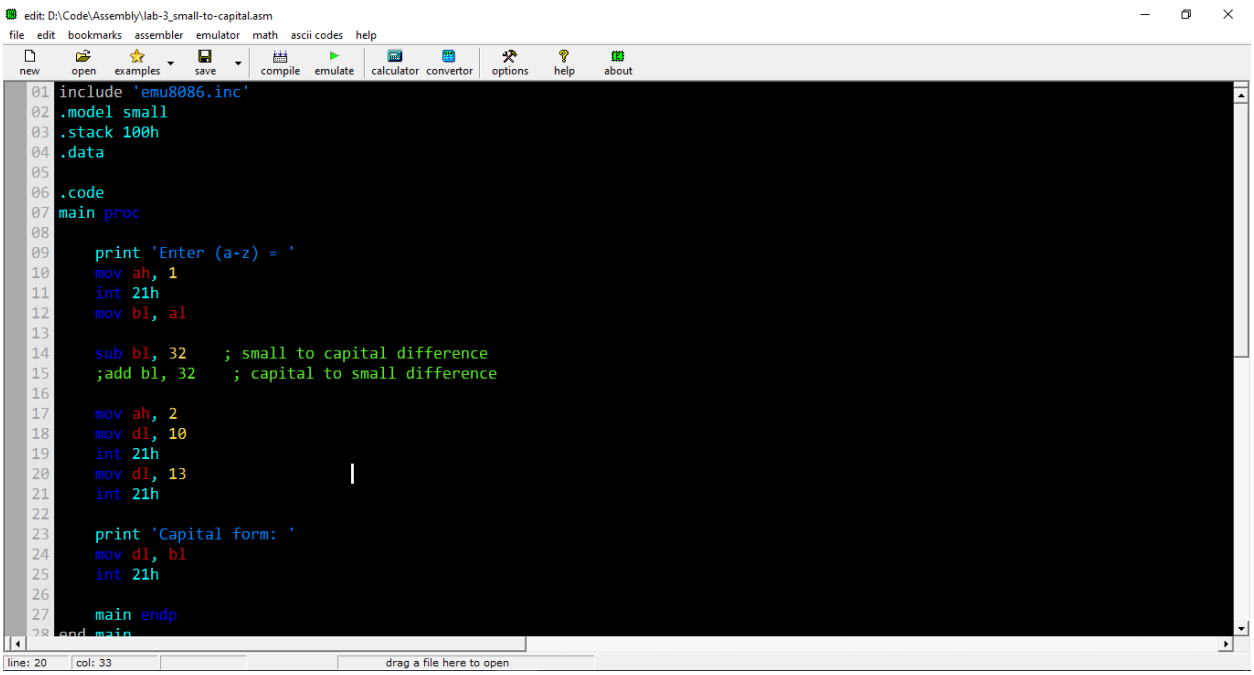
Find maximum/minimum between 2 numbers

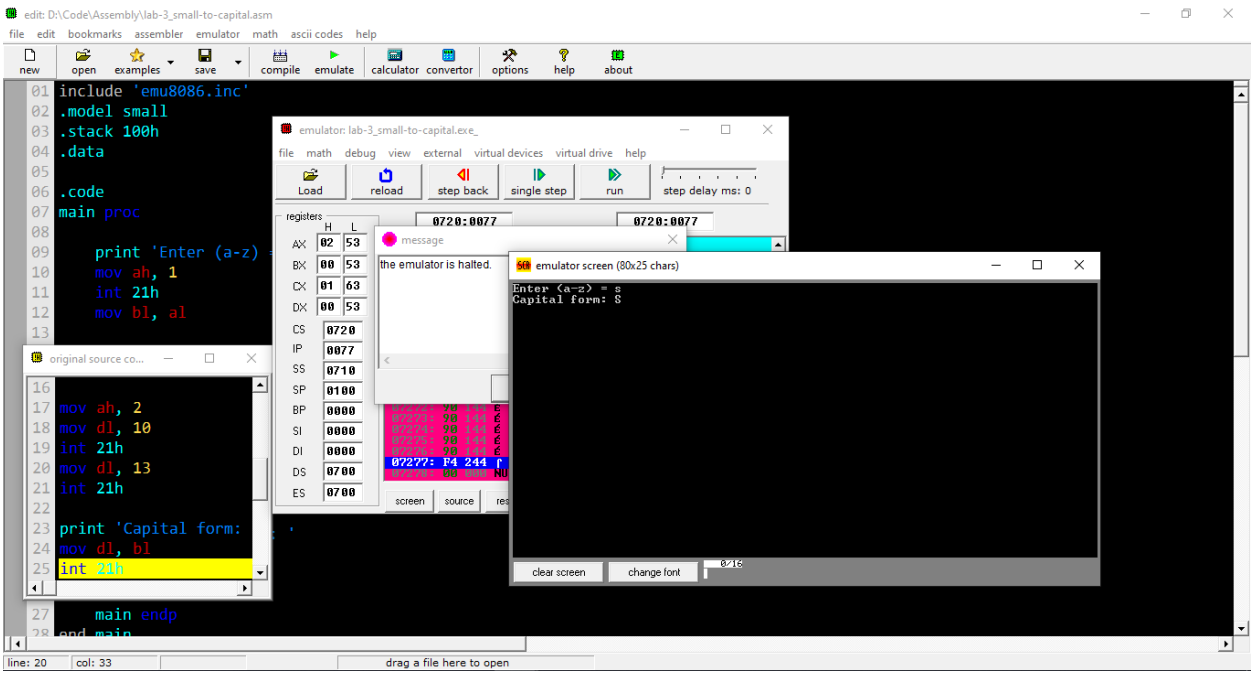




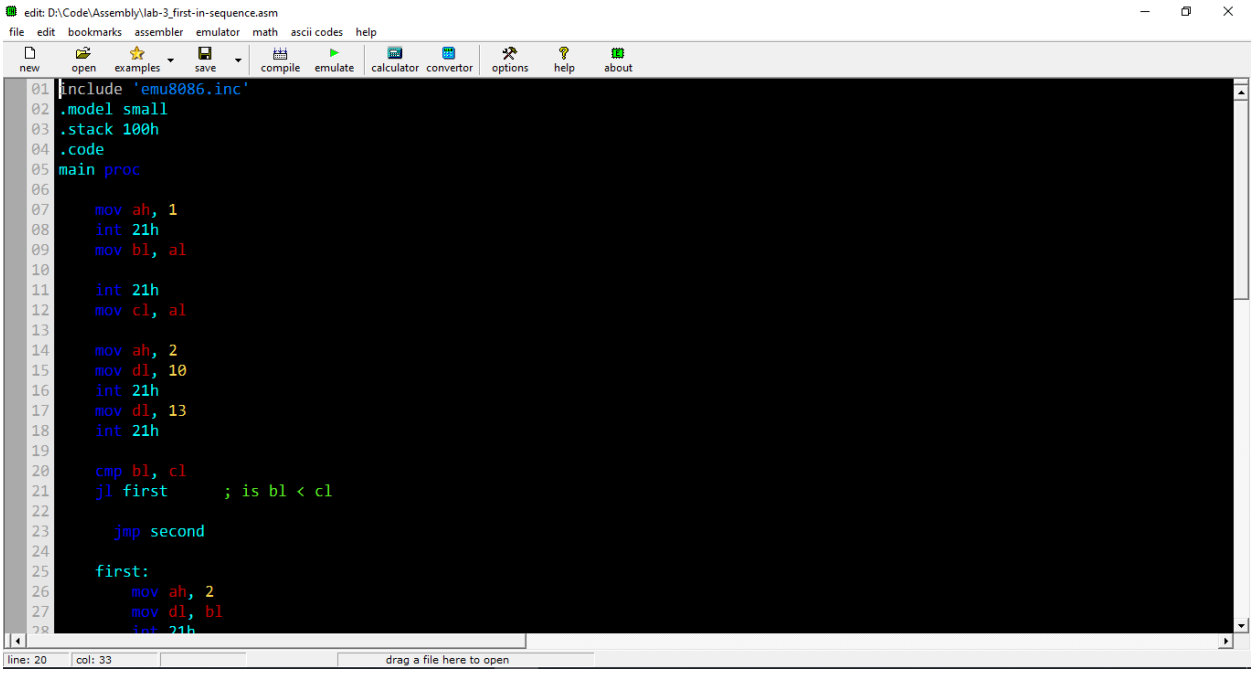


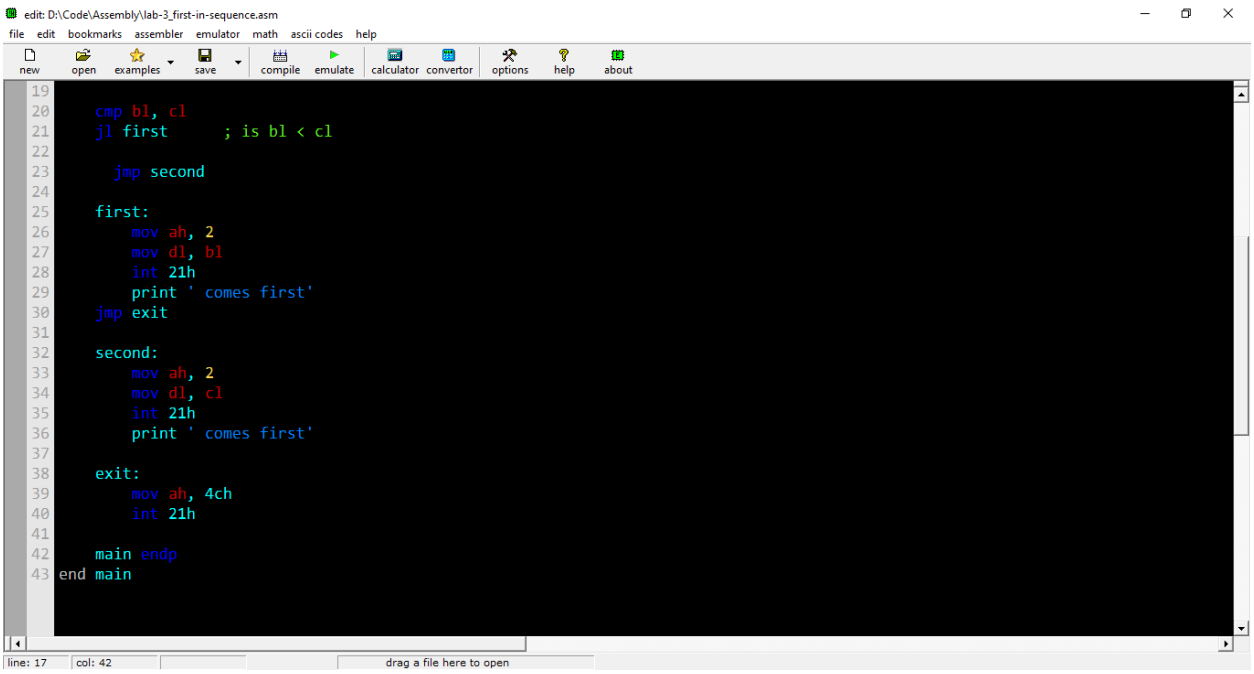
Convert small to capital letter

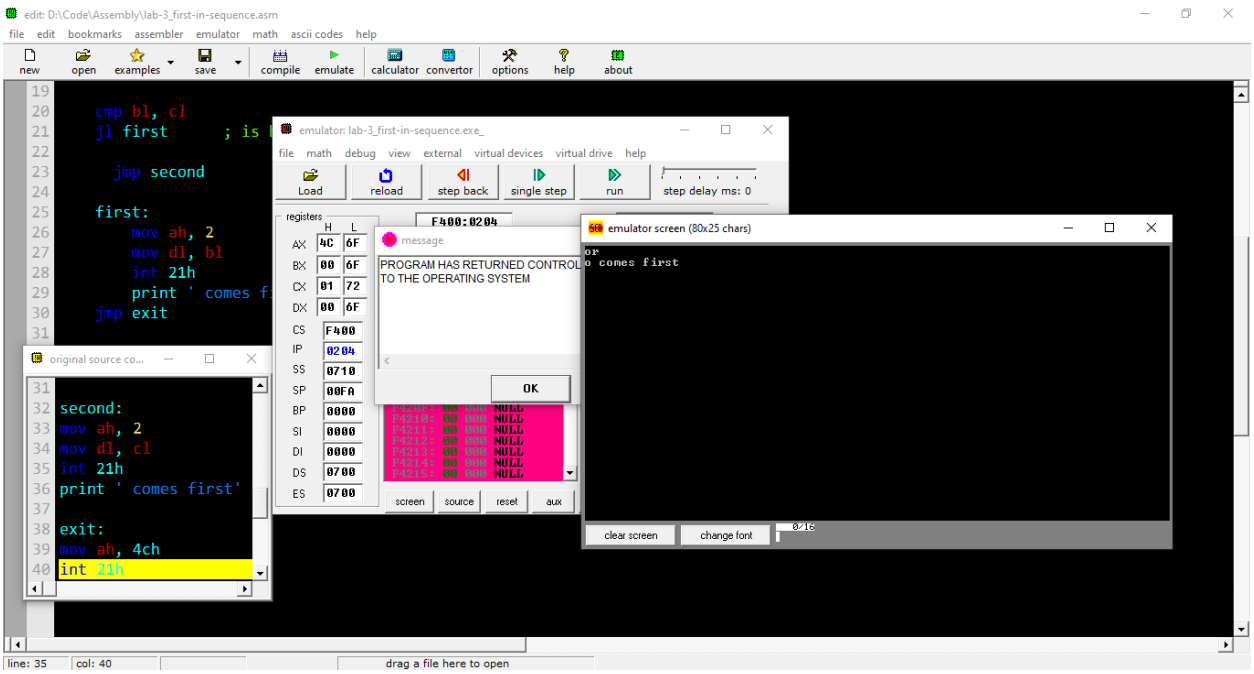




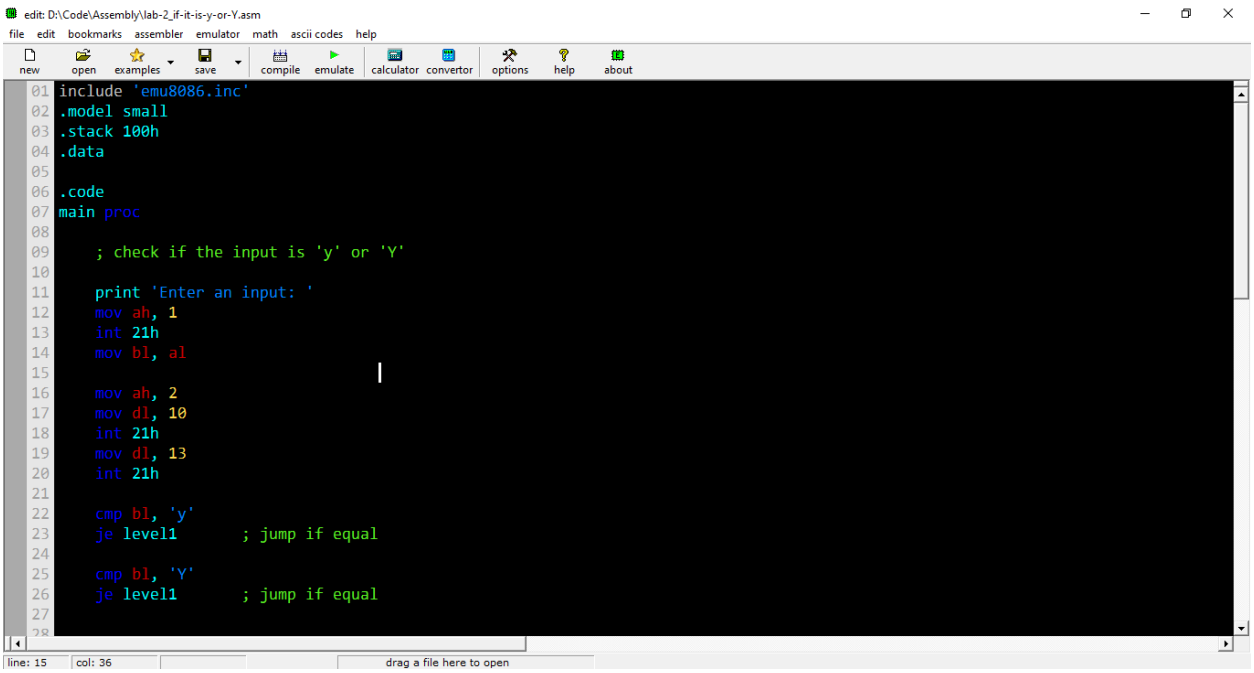
Show which value comes first in sequence between two

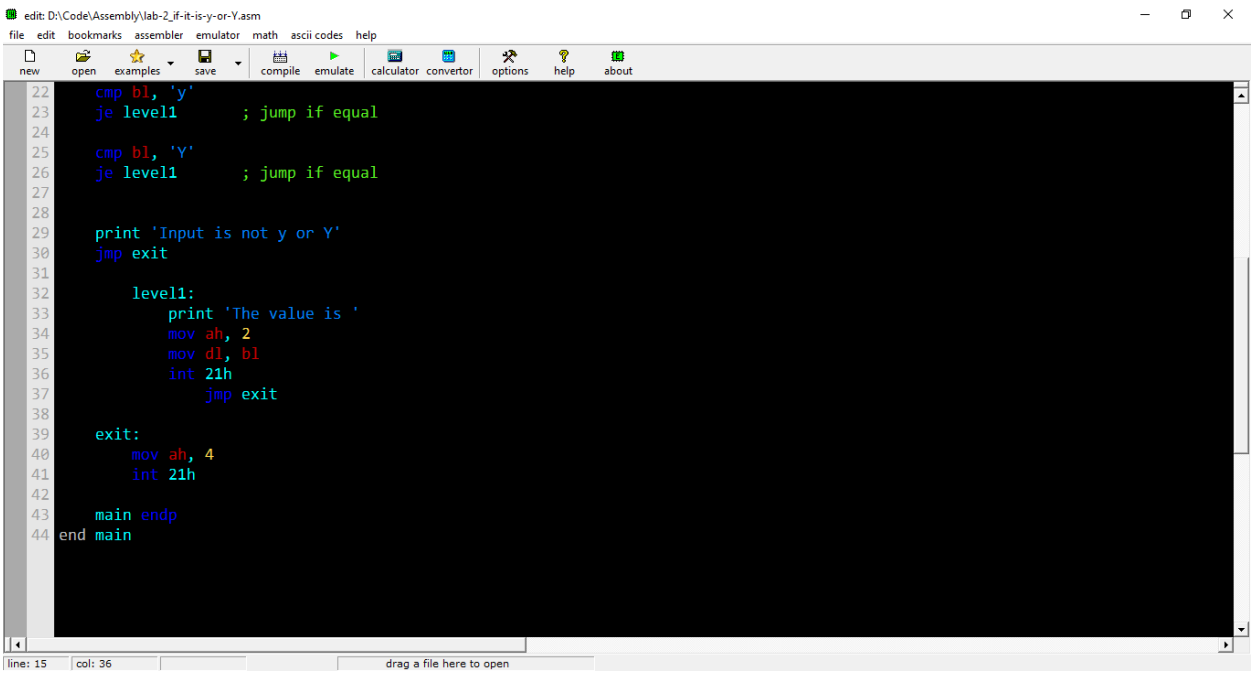

edit: D:\Code\Assembly\lab-3_first-in-sequence.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
01 include 'emu8086.inc'
02 .model small
03 .stack 100h
04 .code
05 main proc
06
07 mov ah, 1
08 int 21h
09 mov 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
line: 20 col: 33
drag a file here to open

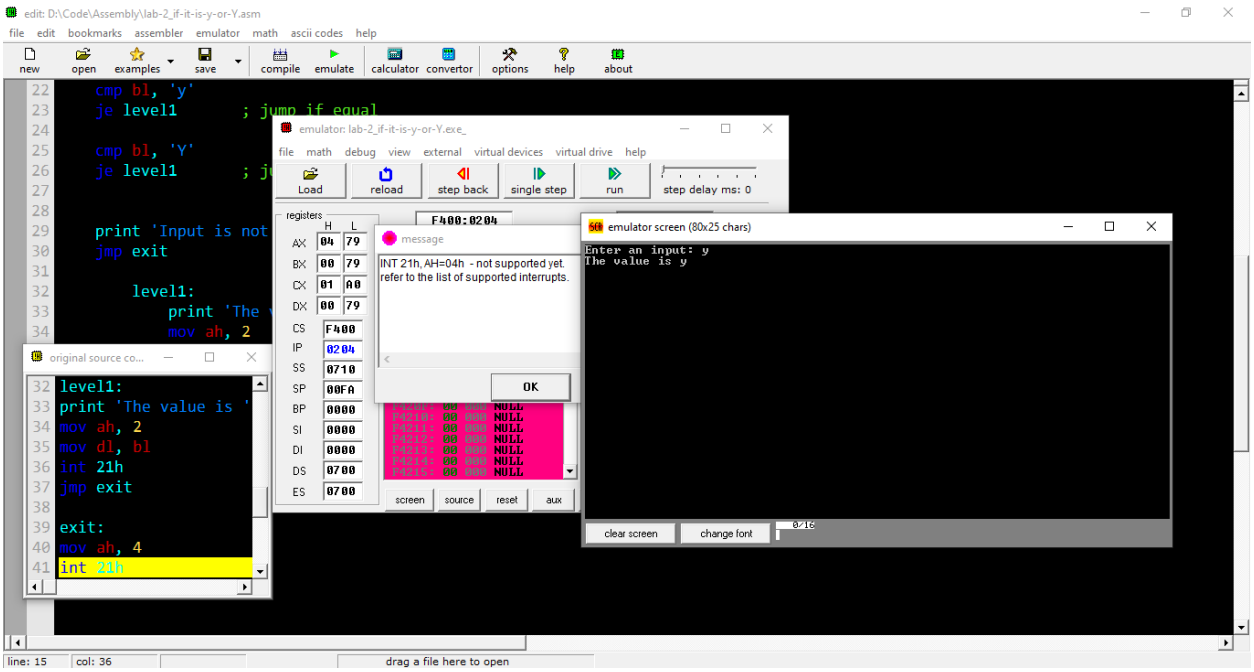

edit: D:\Code\Assembly\lab-3_first-in-sequence.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
19
20 cmp bl, cl
21 jl first ; is bl < cl
22
23 jmp second
24
25 first:
26 mov ah, 2
27 mov dl, bl
28 int 21h
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



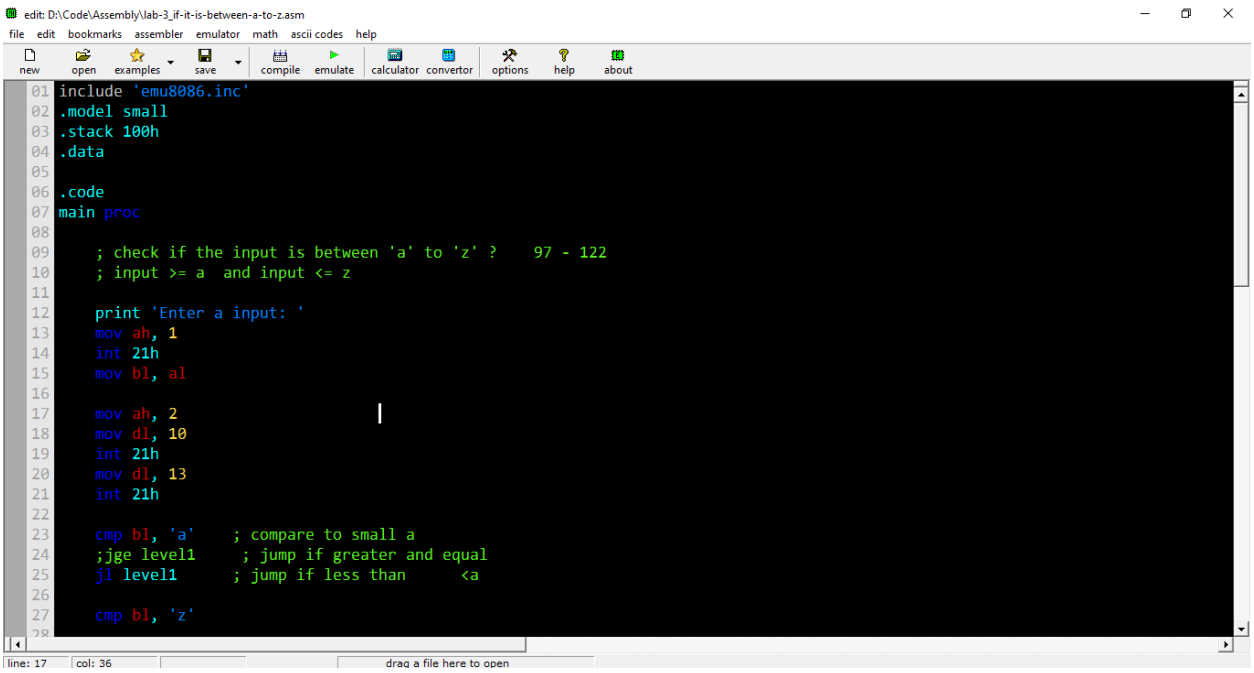
Check if the input is ‘Y’ or ‘y’

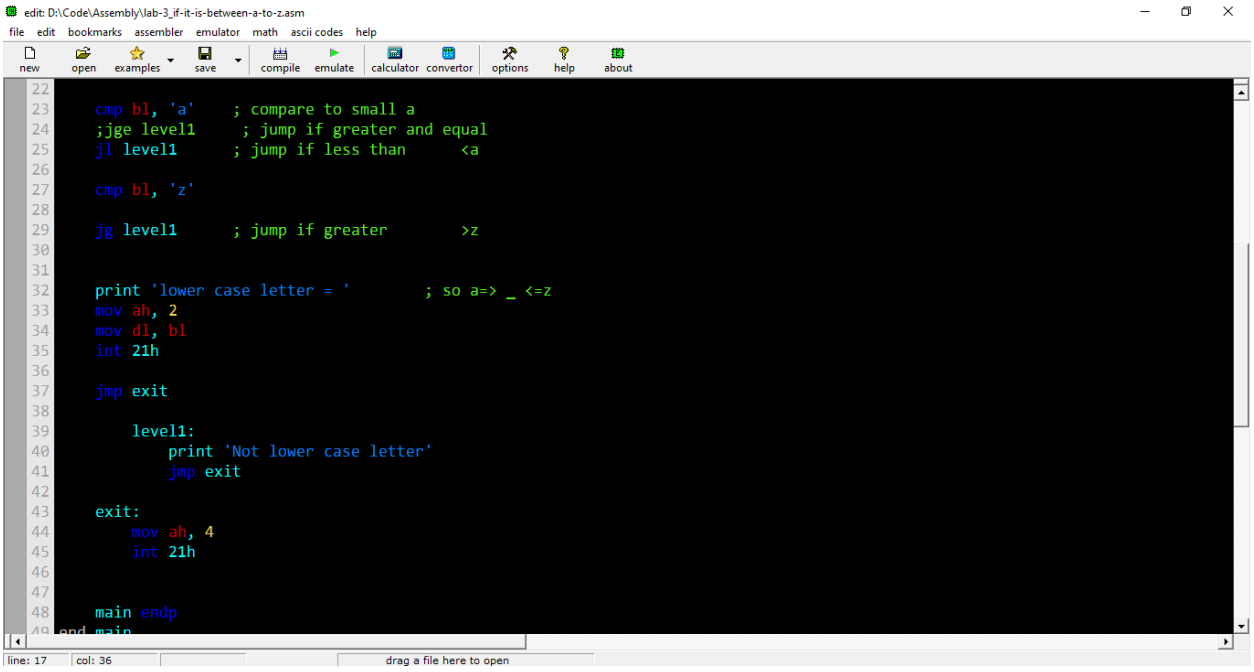


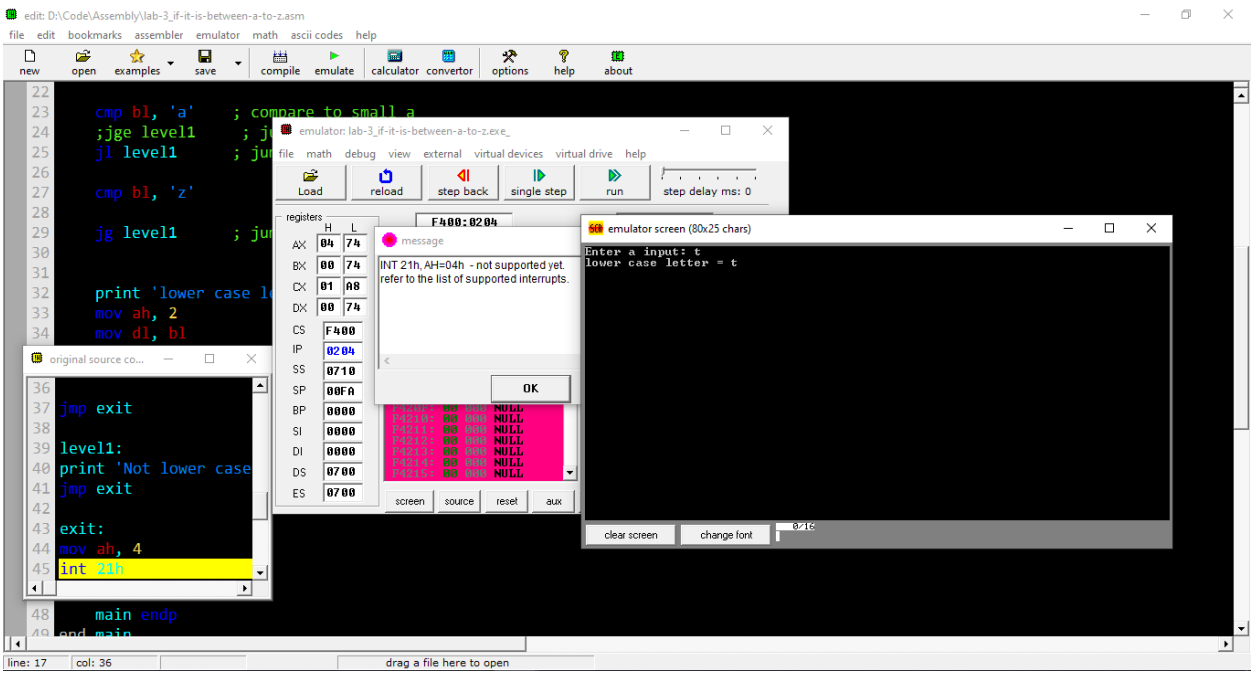




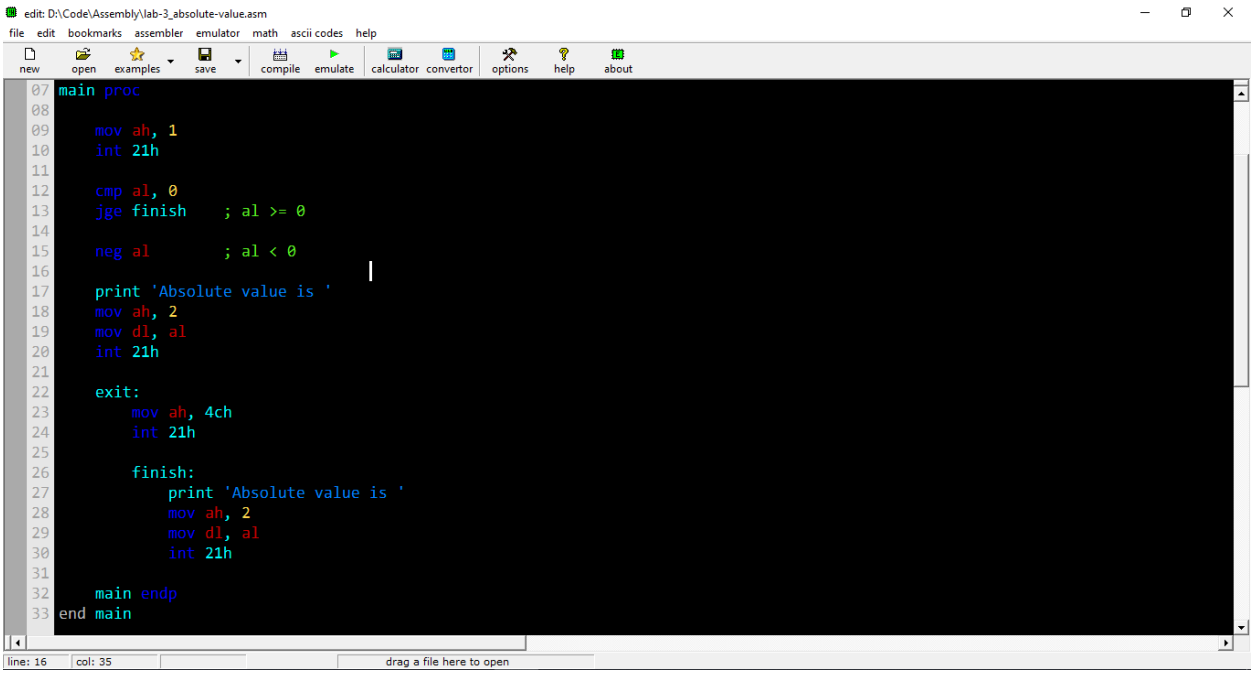
Checking small letter alphabet

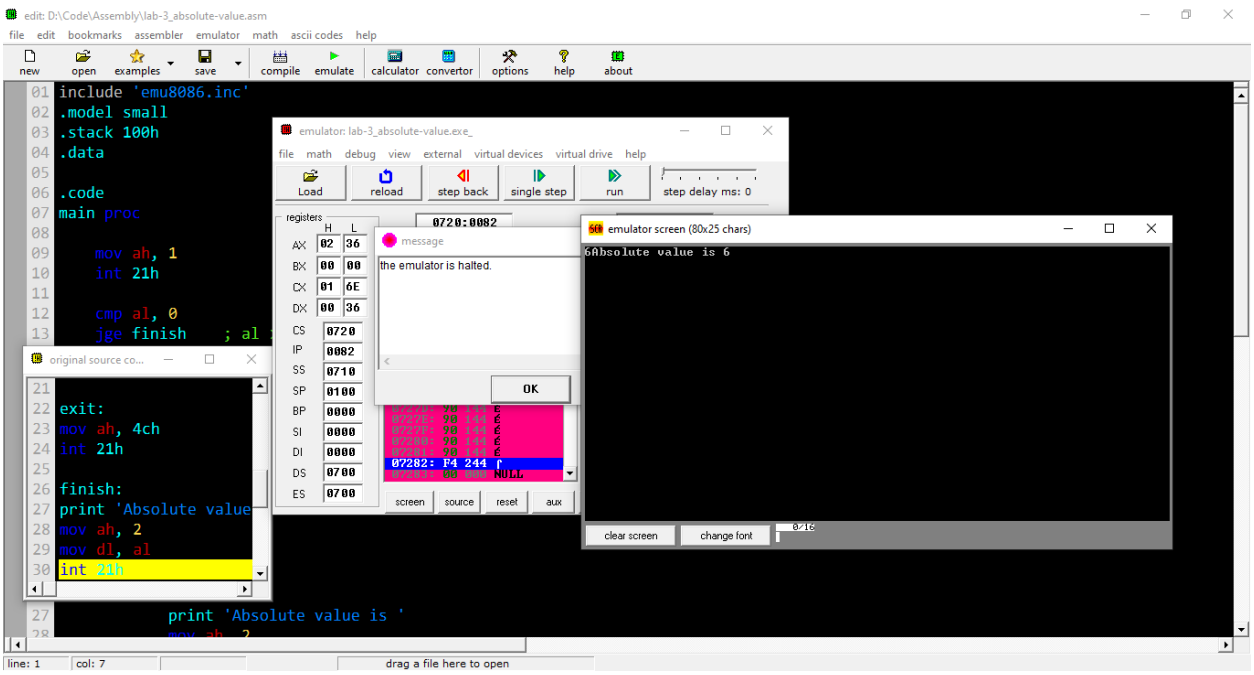

edit: D:\Code\Assembly\lab-3_if-it-is-between-a-to-z.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
include 'emu8086.inc'
.model small
.stack 100h
.data
.code
main proc
; check if the input is between 'a' to 'z' ? 97 - 122
; input >= a and input <= z
print 'Enter a input: '
mov ah, 1
int 21h
mov bl, al
mov ah, 2
mov dl, 10
int 21h
mov dl, 13
int 21h
cmp bl, 'a' ; compare to small a
;jge level1 ; jump if greater and equal
jl level1 ; jump if less than <a
cmp bl, 'z'
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
cmp bl, 'a' ; compare to small a
;jge level1 ; jump if greater and equal
jl level1 ; jump if less than <a
cmp bl, 'z'
jg level1 ; jump if greater >z
print 'lower case letter = ' ; so a=> _ <=z
mov ah, 2
mov dl, bl
int 21h
jmp exit
level1:
print 'Not lower case letter'
jmp exit
exit:
mov ah, 4
int 21h
main endp
end main
line: 17 col: 36
drag a file here to open

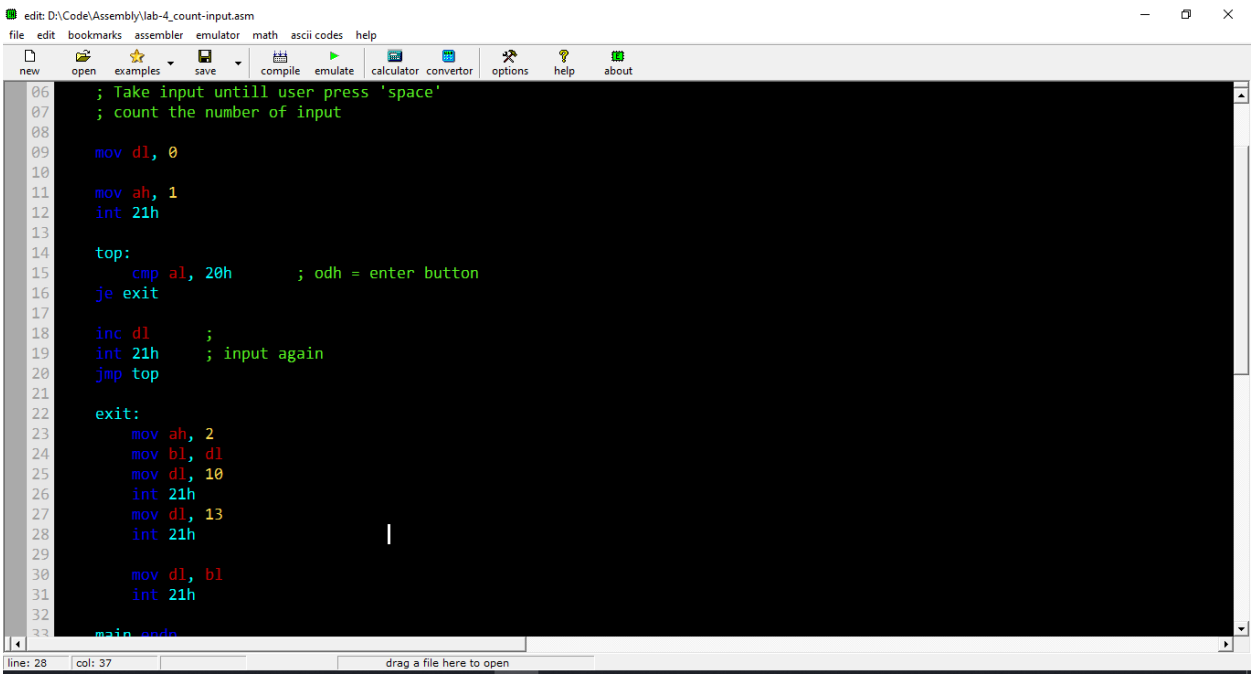


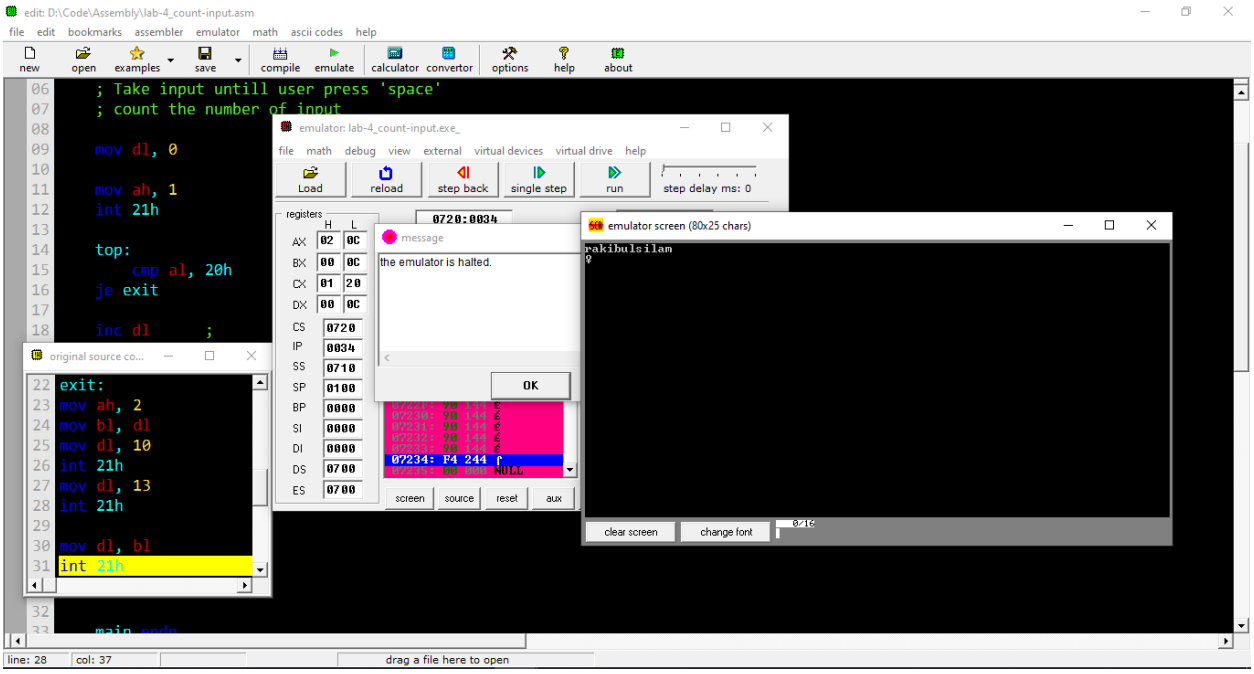
Find the absolute value



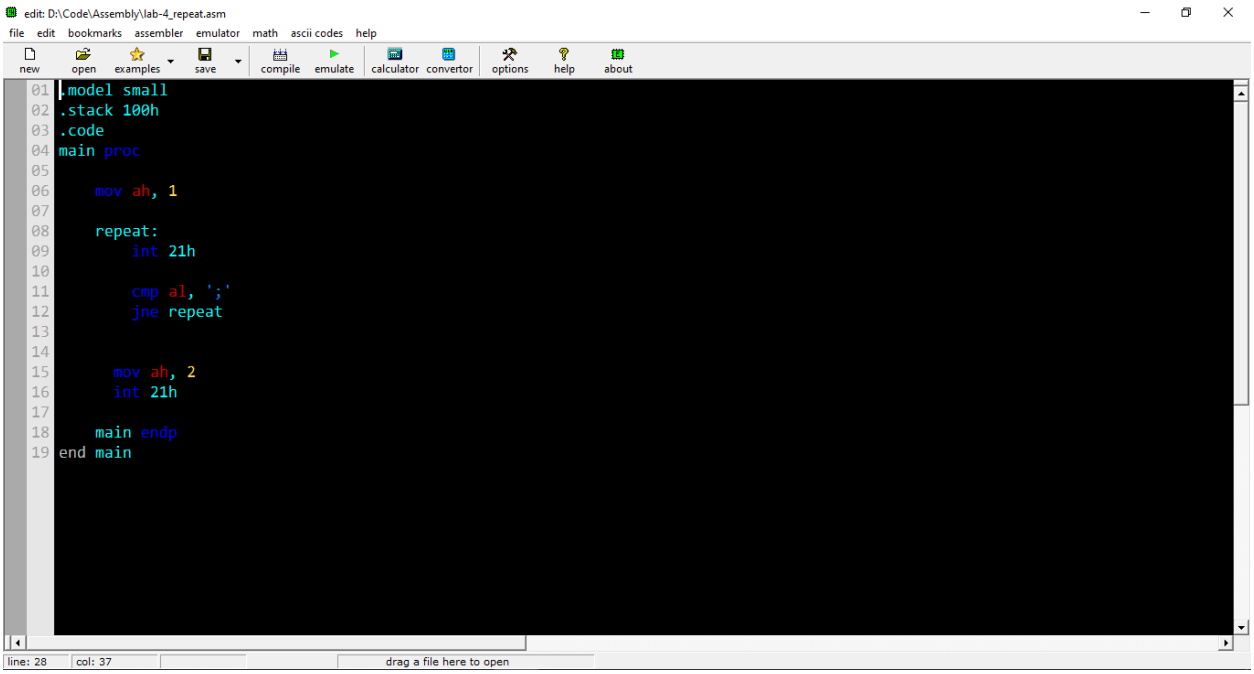


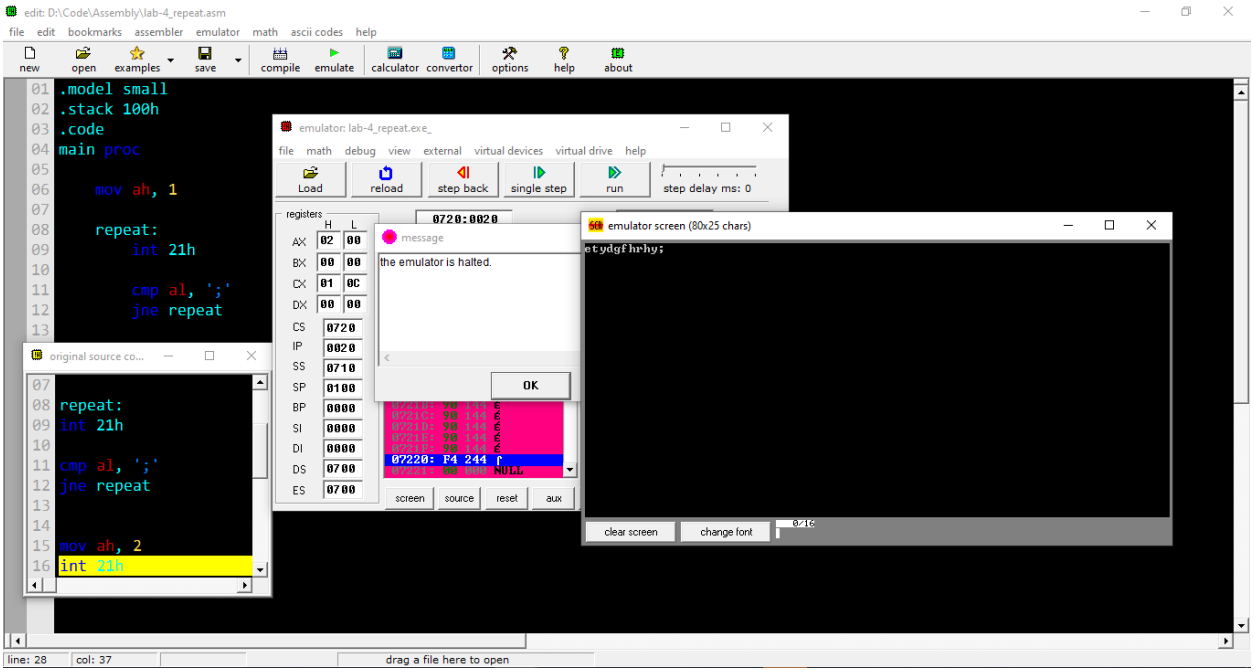
Count input until the space is given



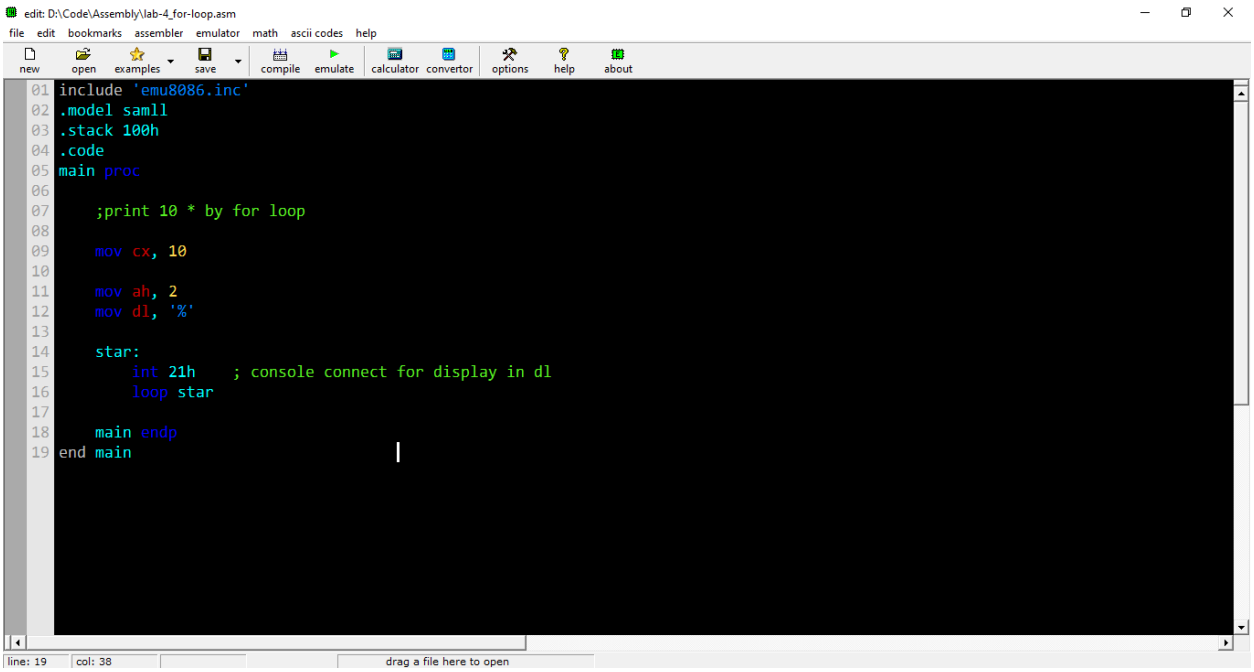


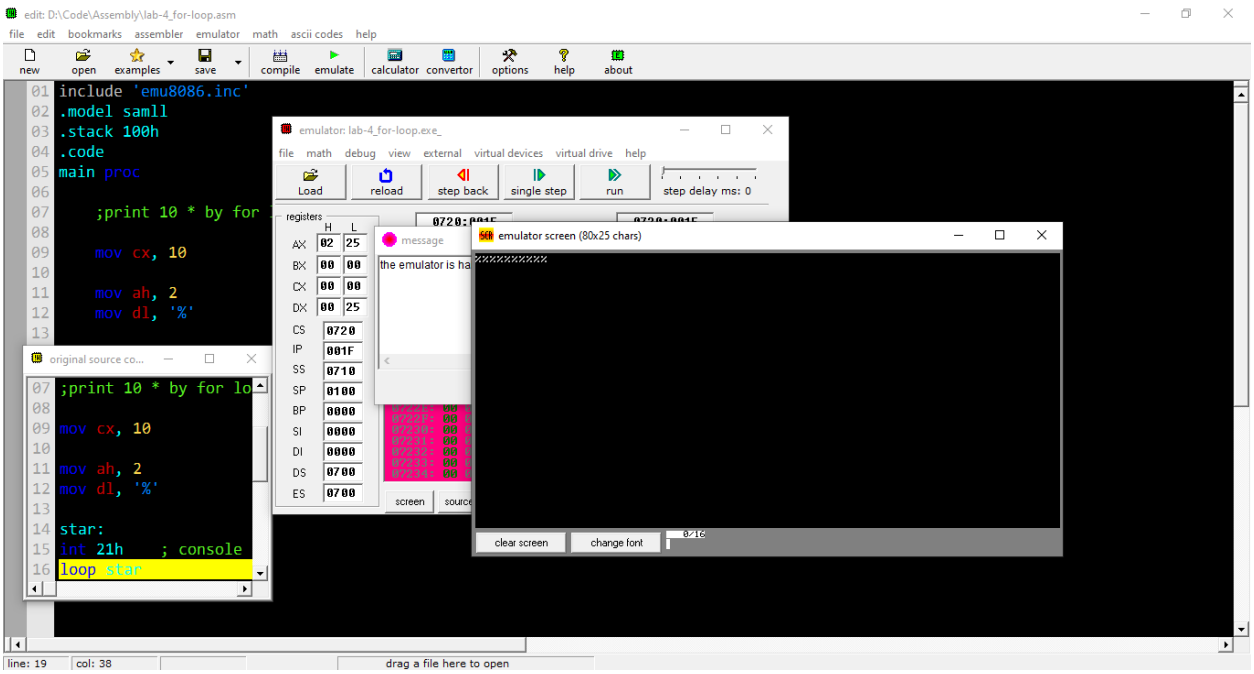
Use of repeat



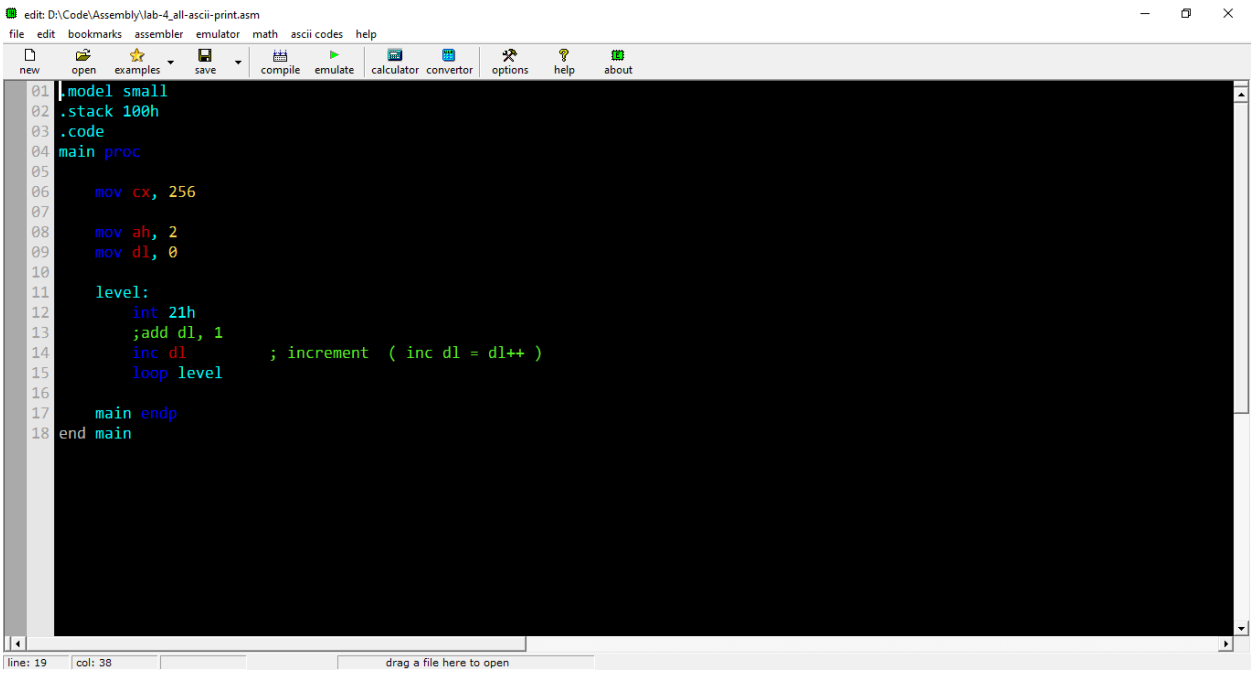


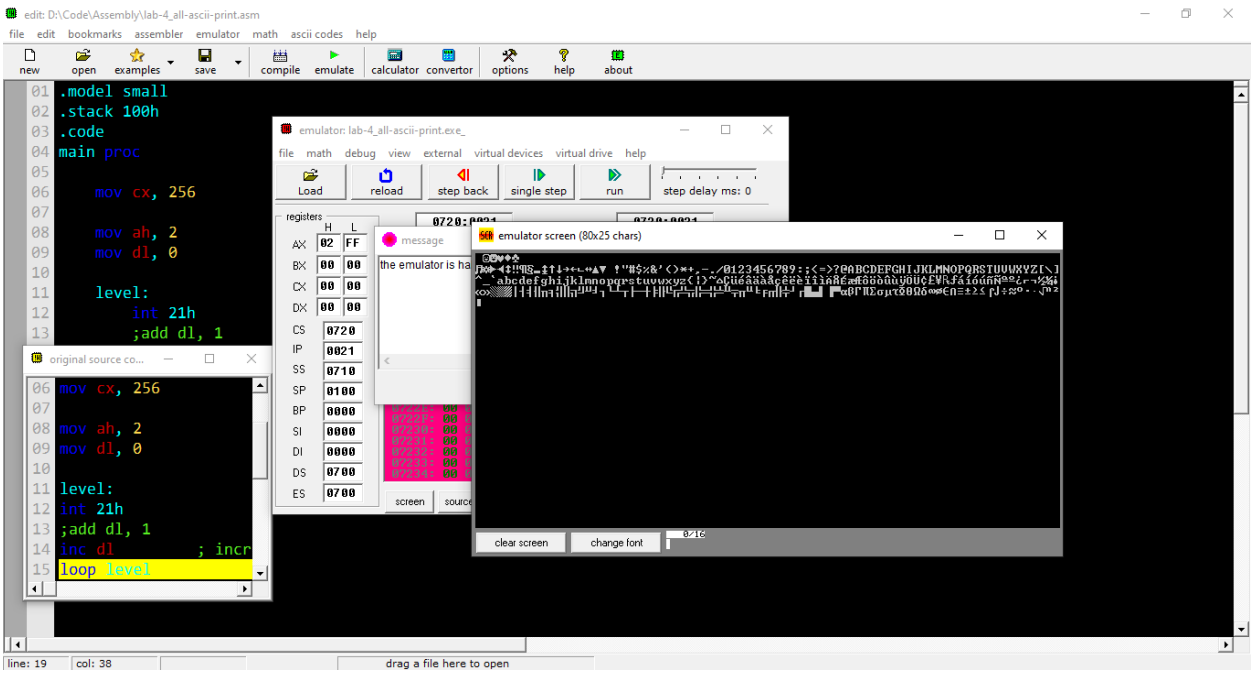
Print 10 ‘%’ by using for loop



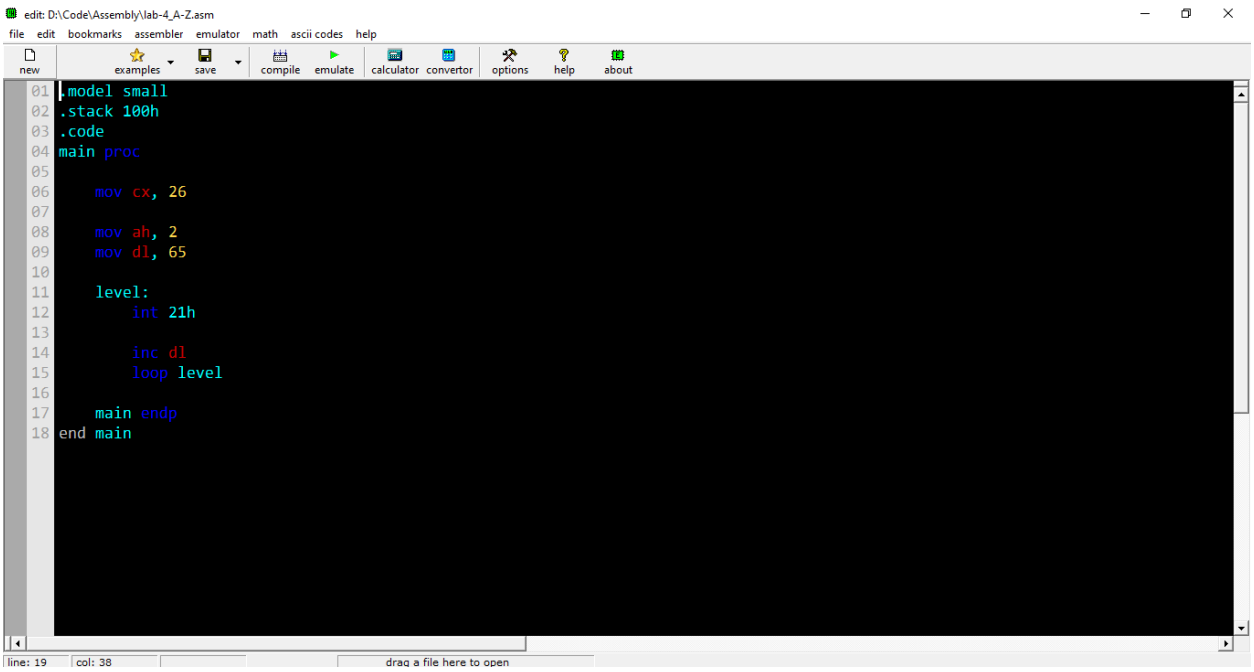


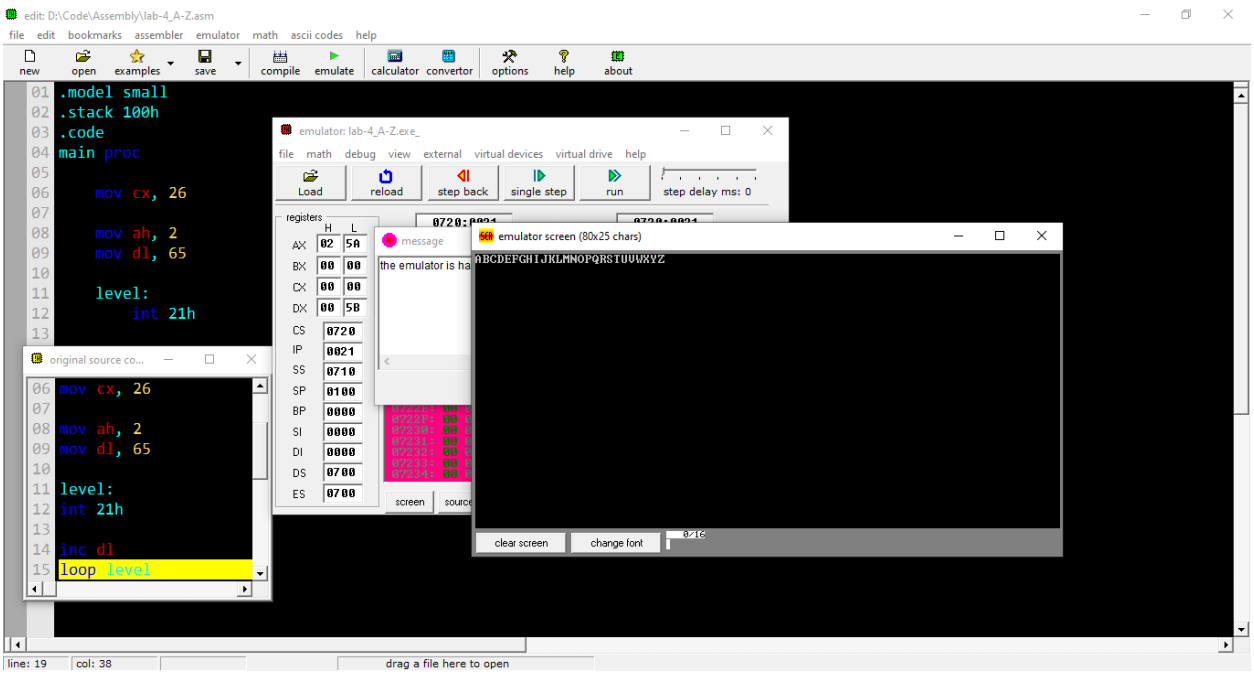
Printing all ascii values



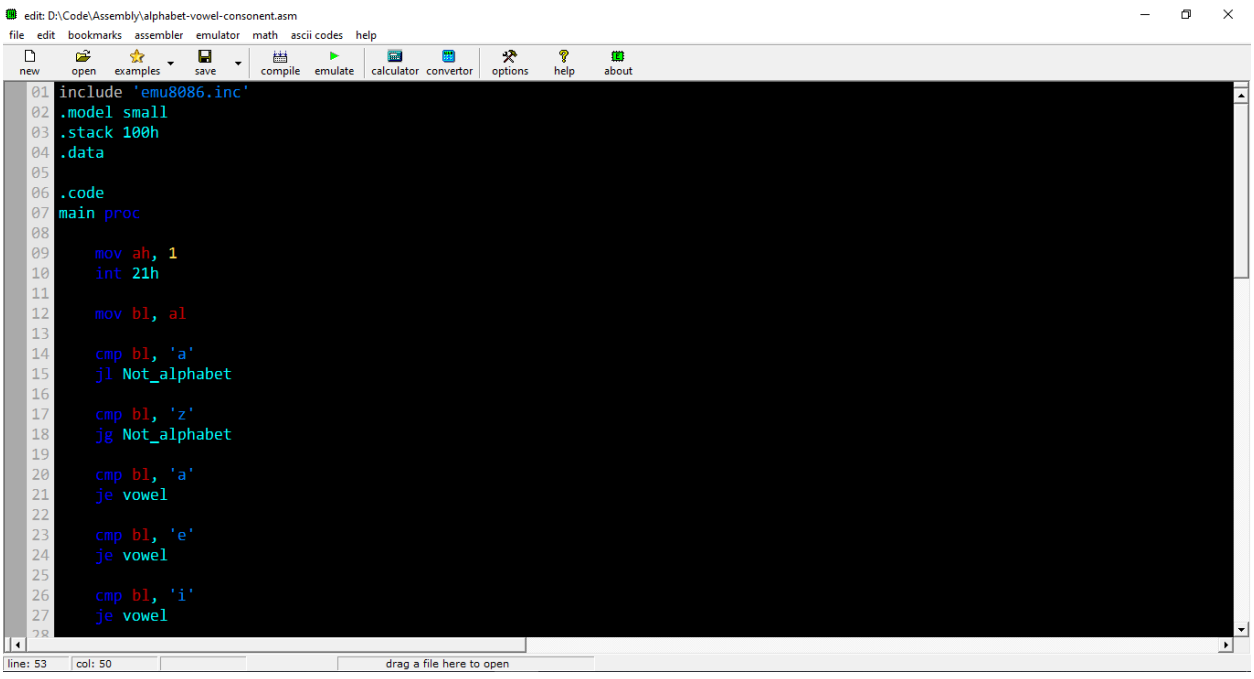


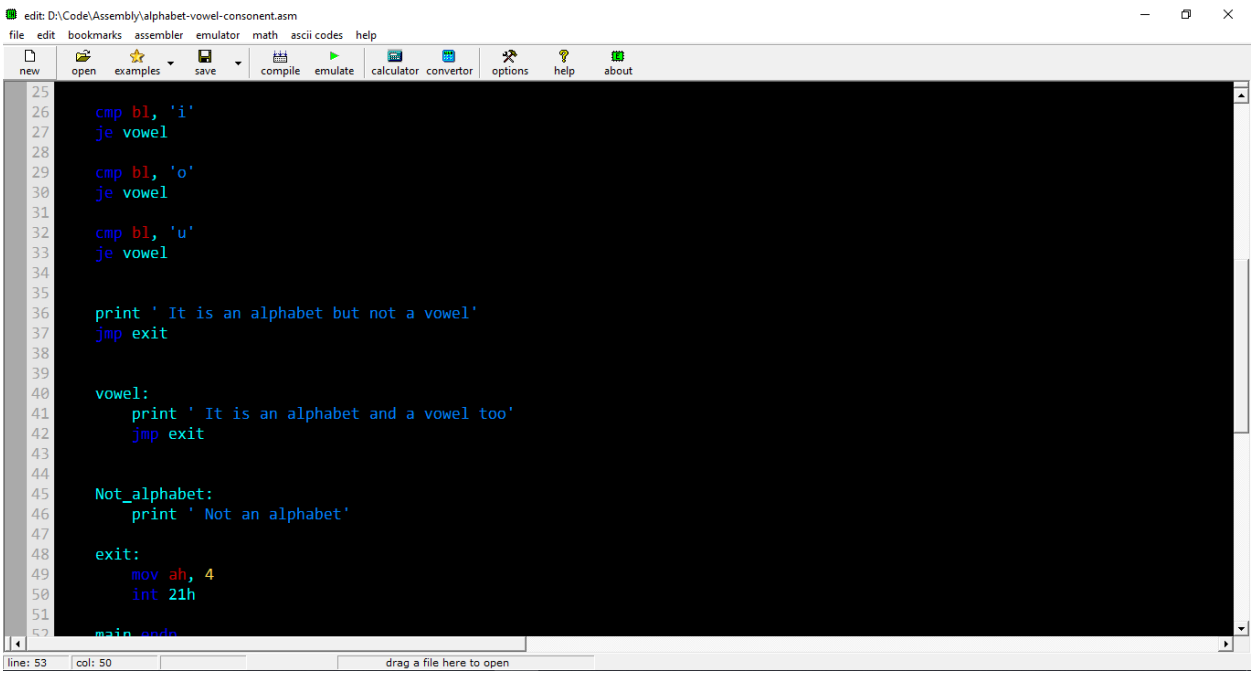
Print capital letters

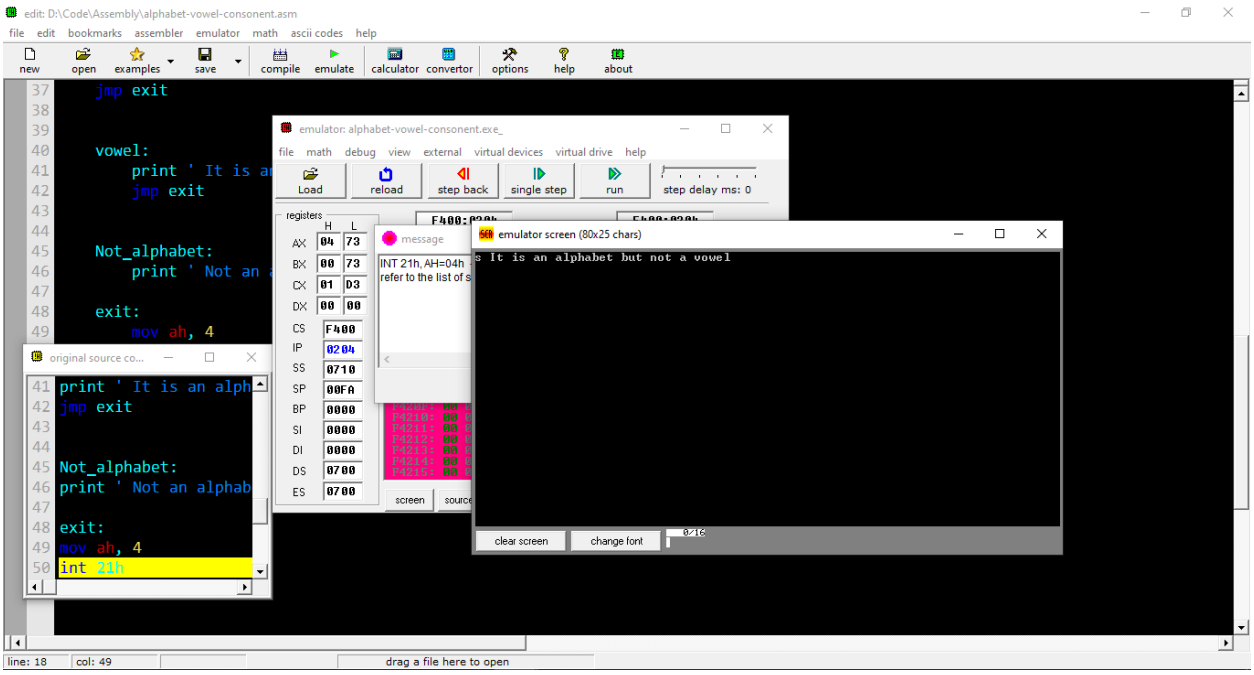




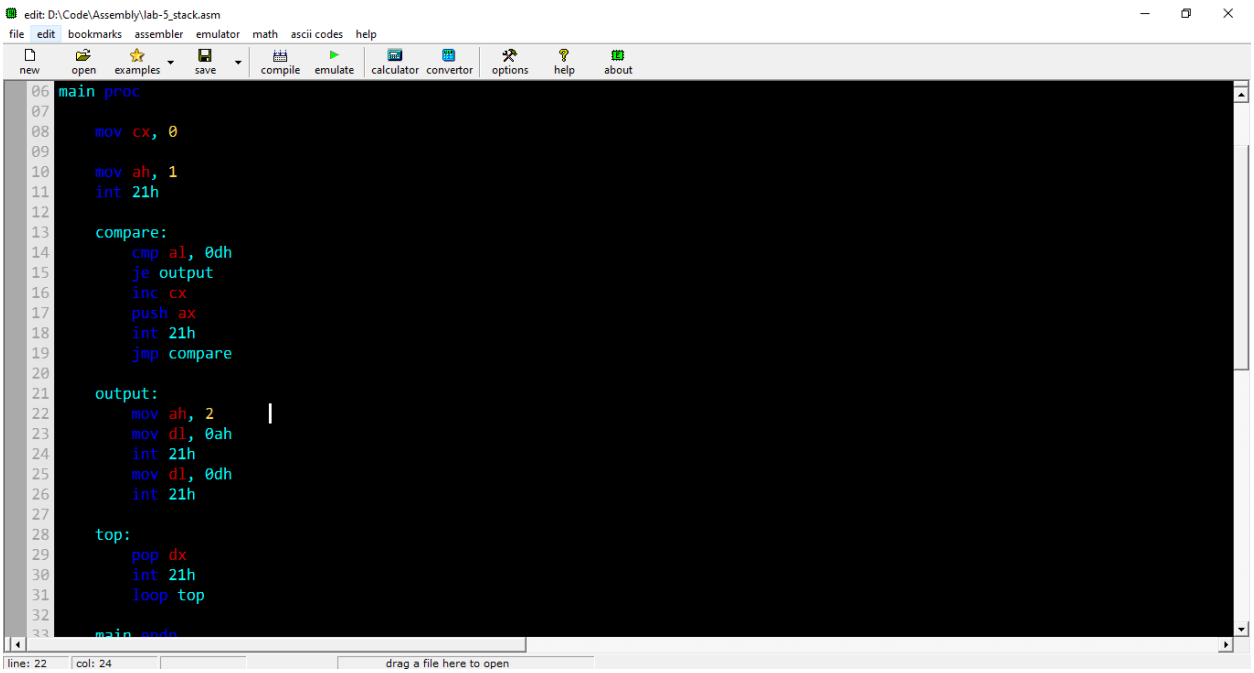
Check an input is alphabet or vowel or anything else

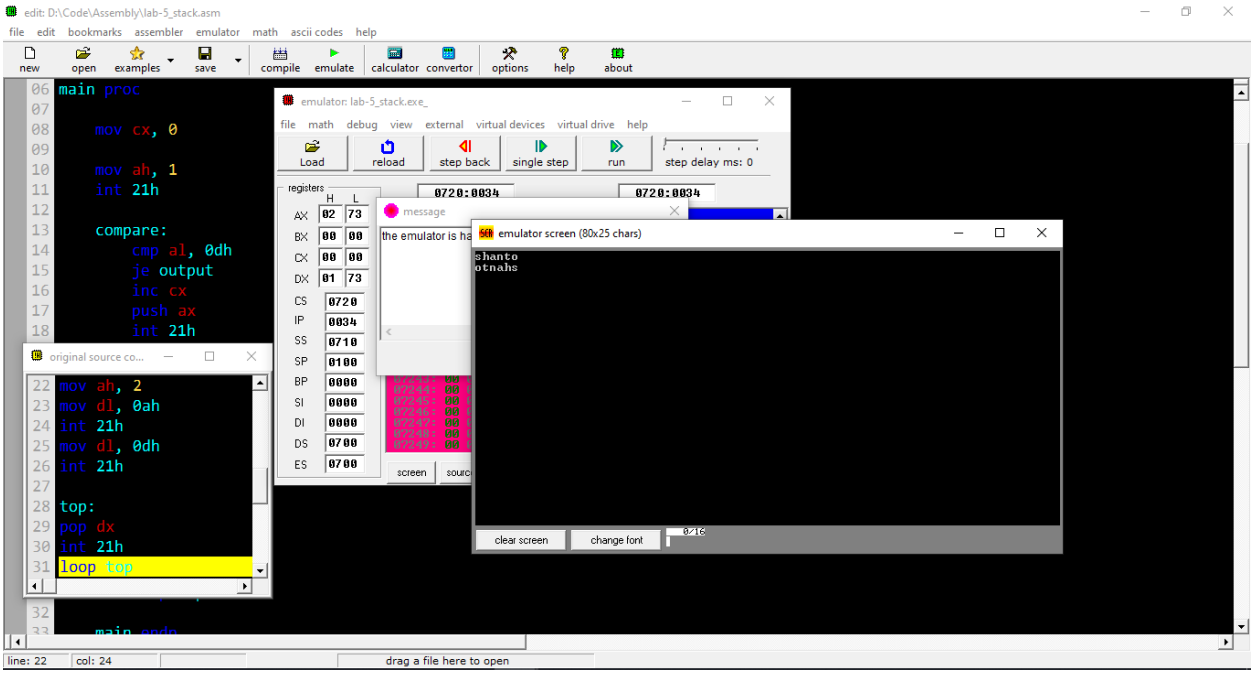




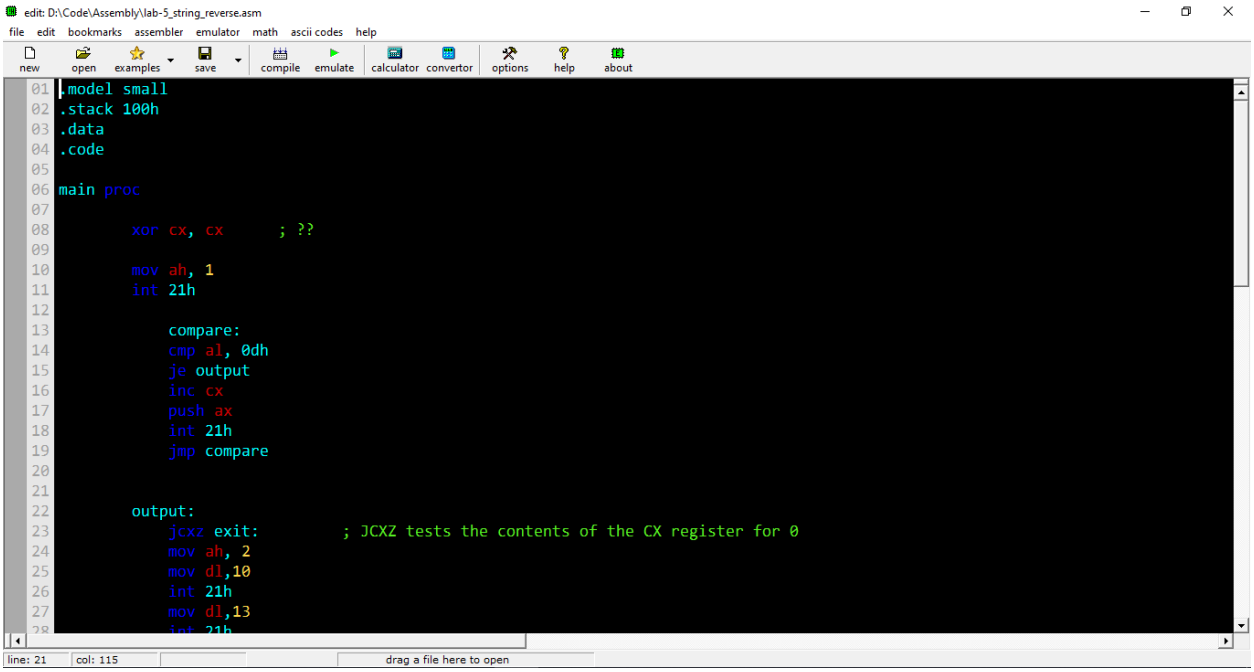


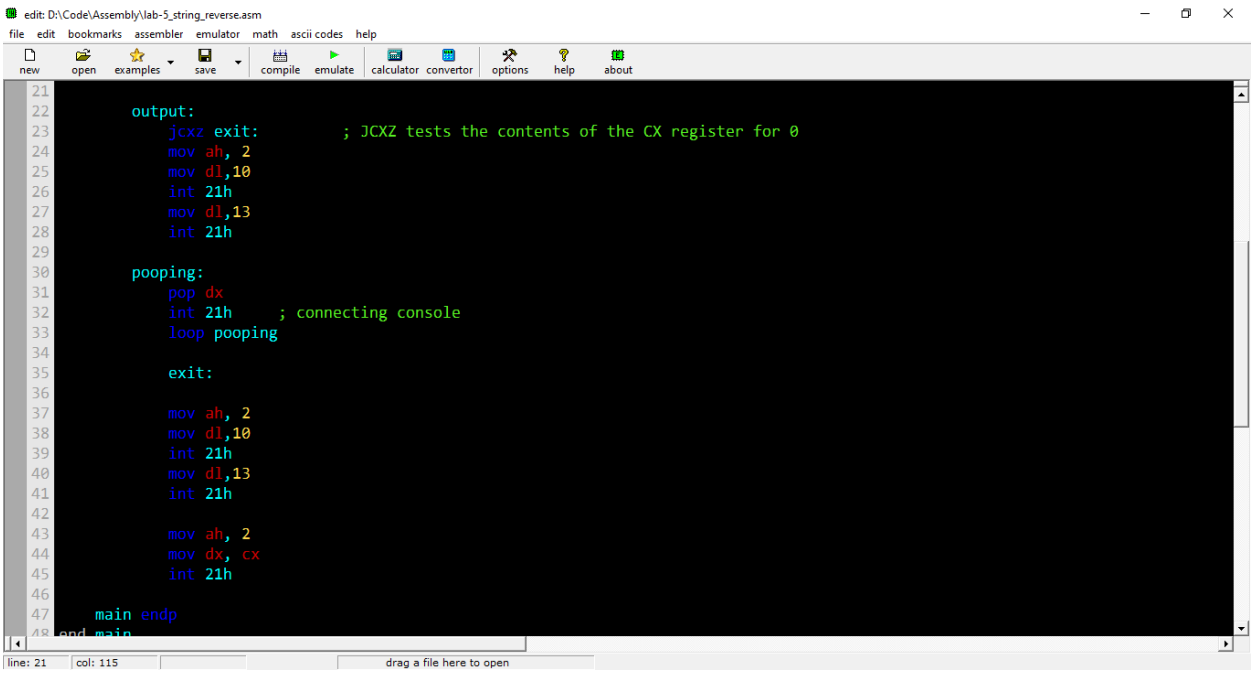
Take some inputs and show the outputs by stack

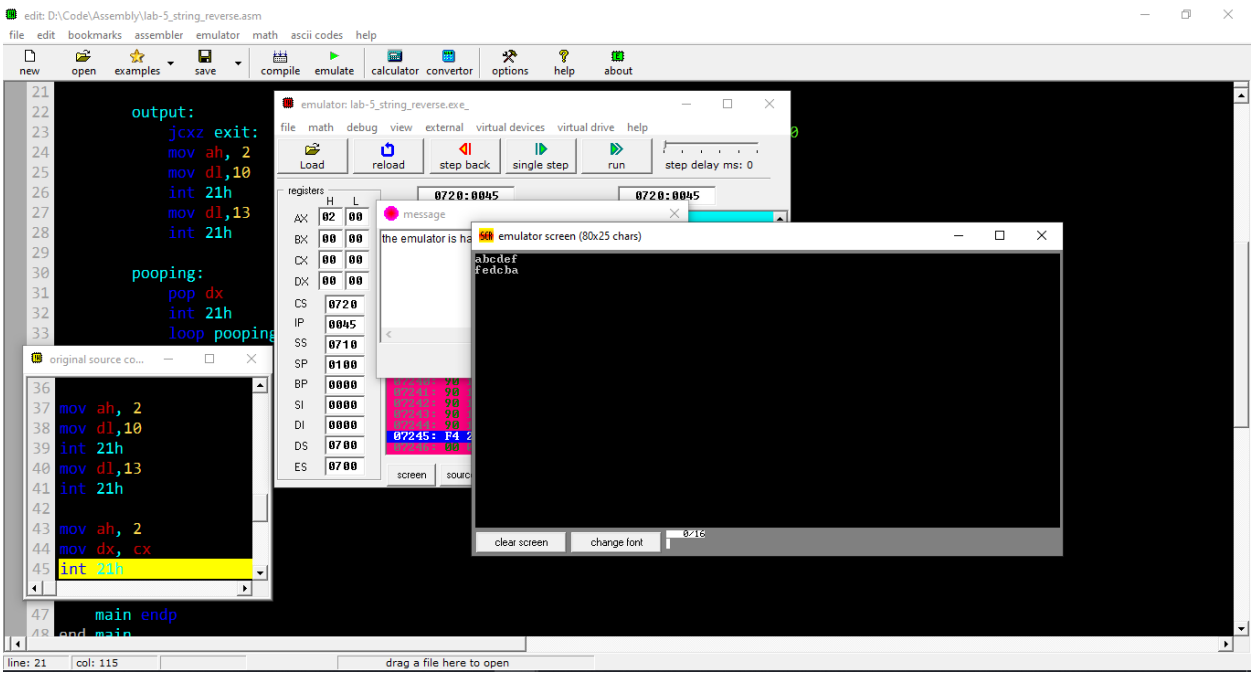




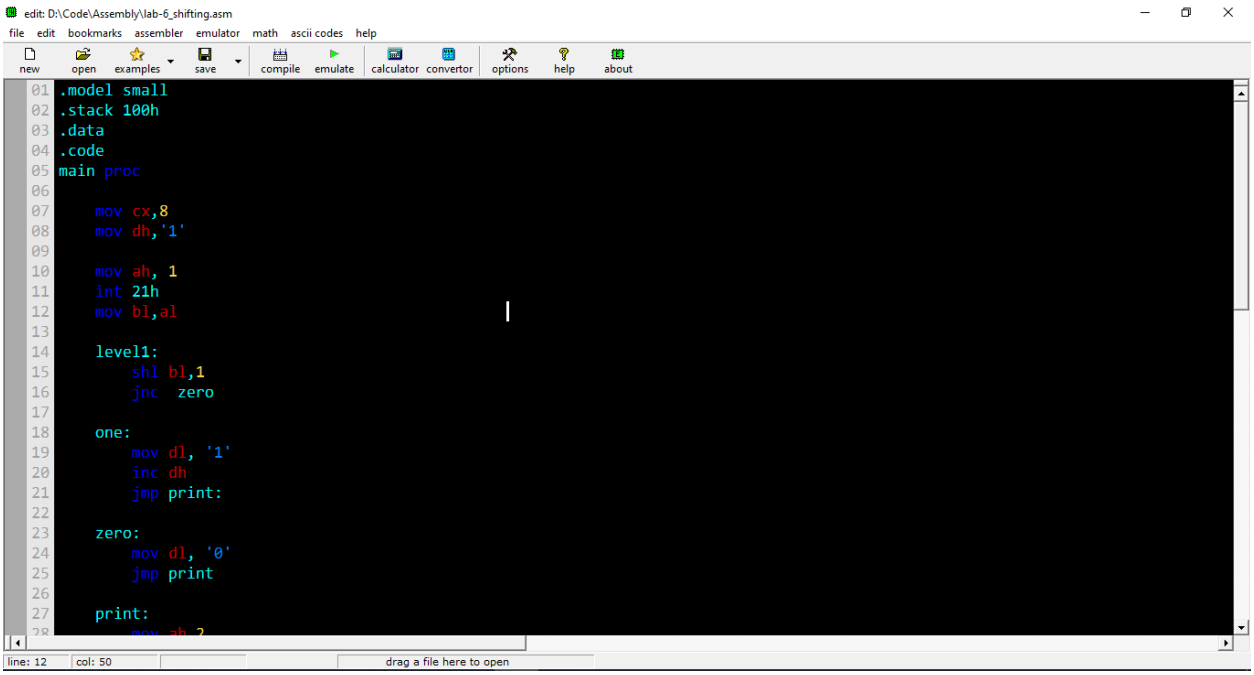
Reversing string by push and pop

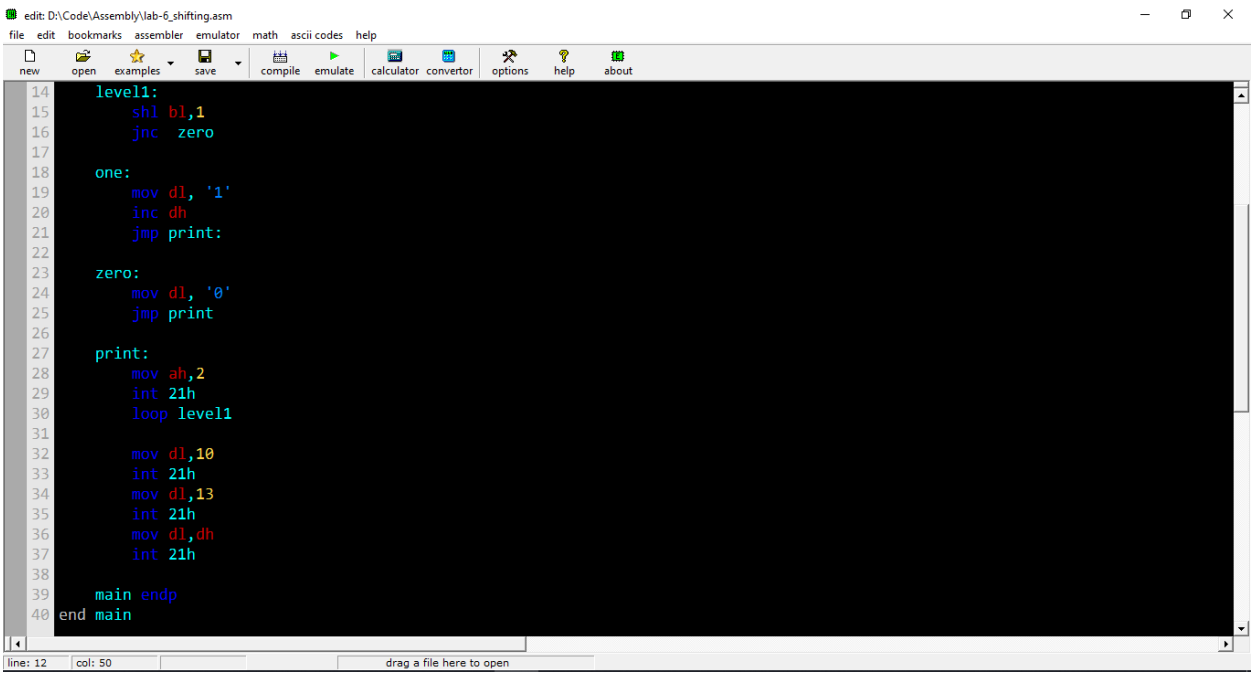

edit: D:\Code\Assembly\lab-5_string_reverse.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
01 .model small
02 .stack 100h
03 .data
04 .code
05
06 main proc
07
08 xor cx, cx ; ??
09
10 mov ah, 1
11 int 21h
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
int 21h
line: 21 col: 115
drag a file here to open

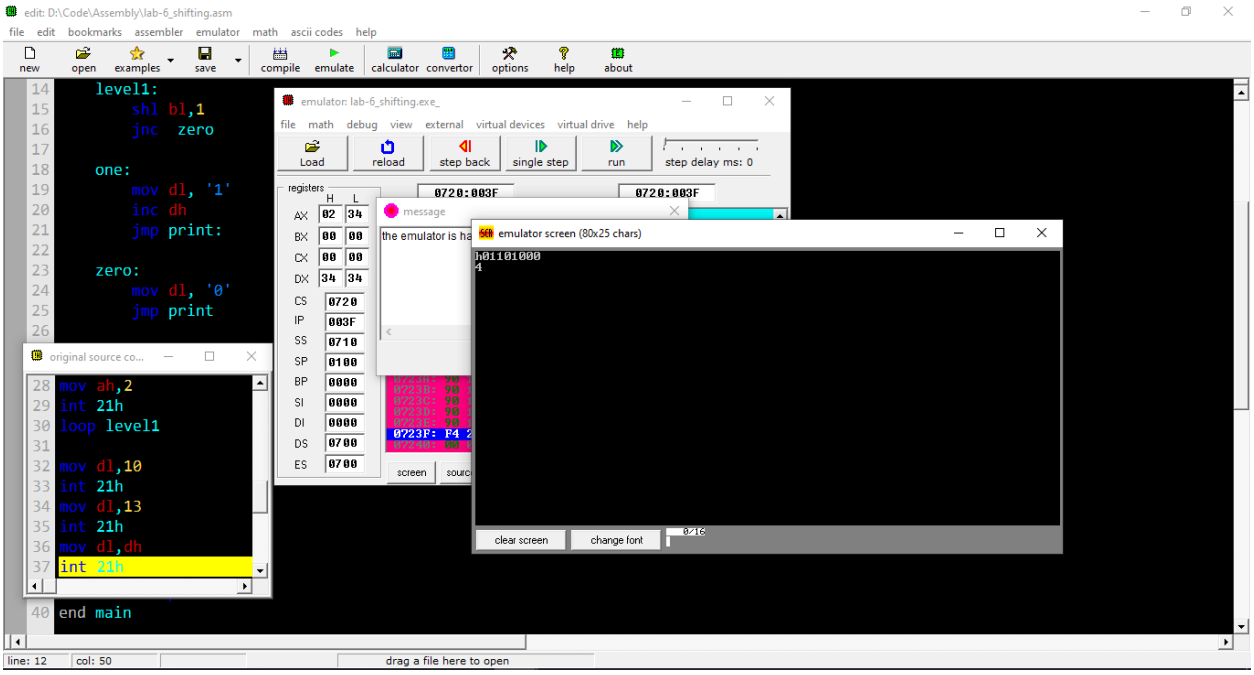

edit: D:\Code\Assembly\lab-5_string_reverse.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
22 output:
23 jcxz exit: ; JCXZ tests the contents of the CX register for 0
24 mov ah, 2
25 mov dl,10
26 int 21h
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
line: 21 col: 115
drag a file here to open



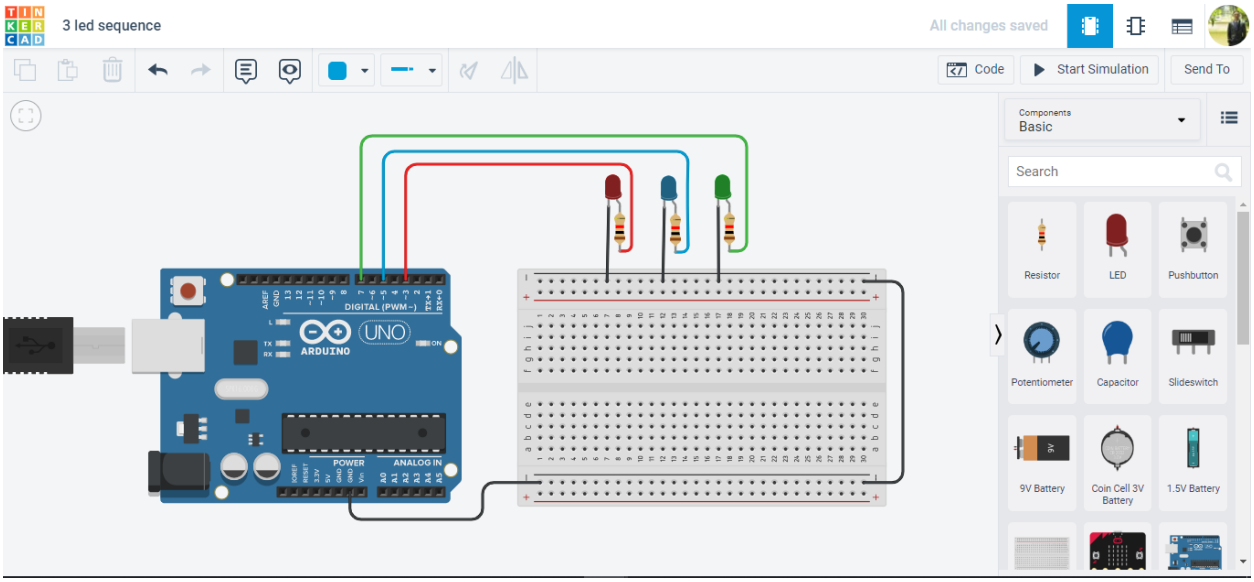
Show shifting by assembly

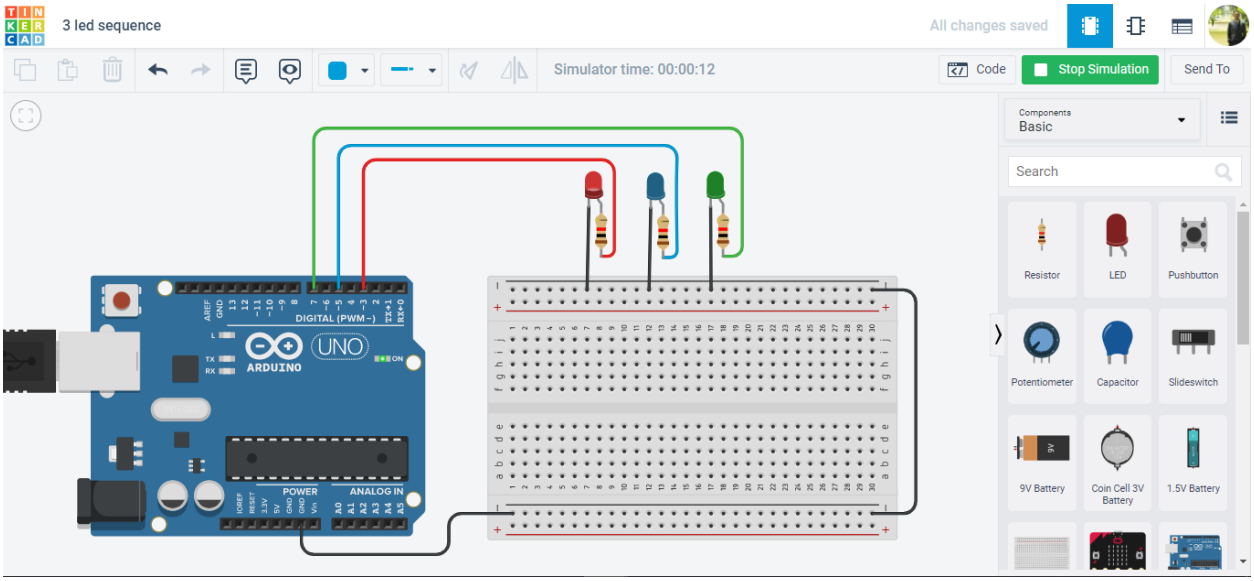


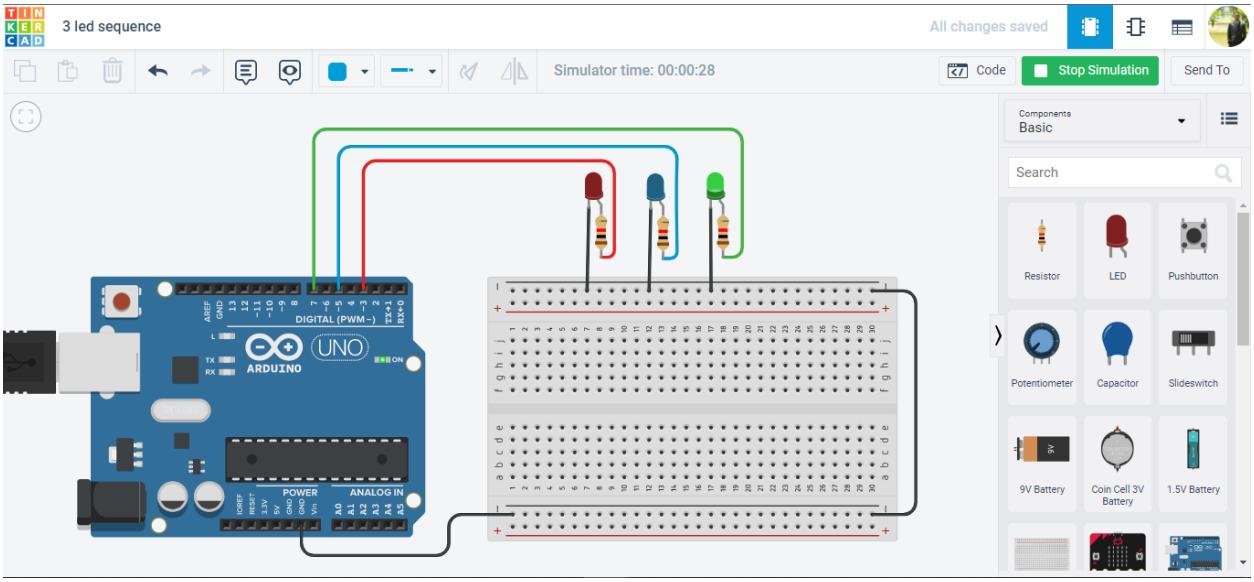




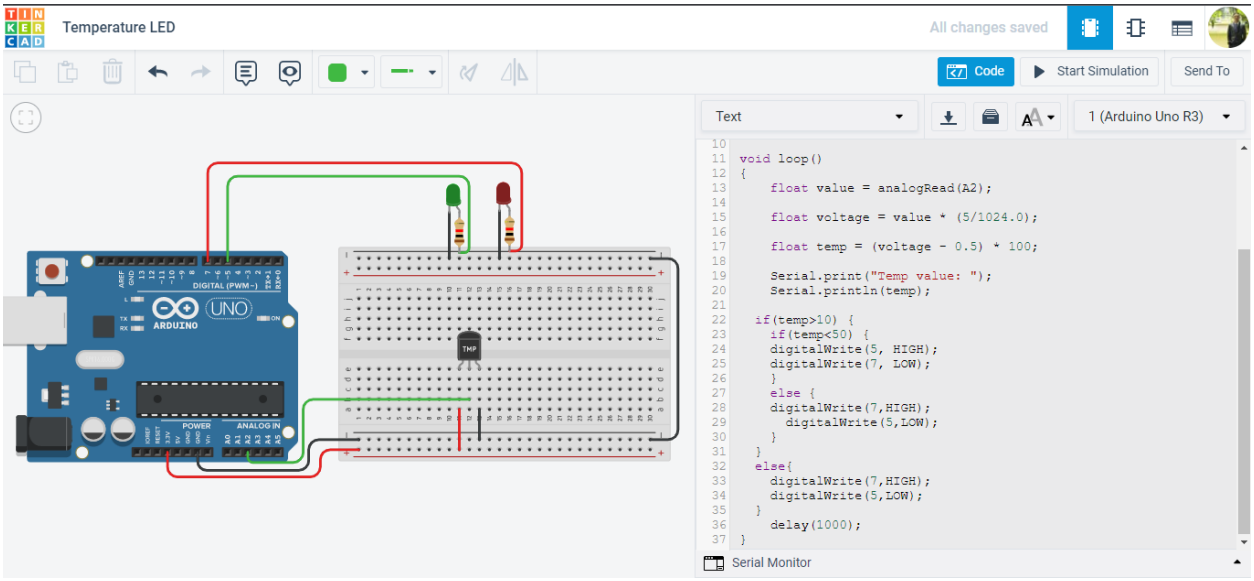
3 sequences of LED

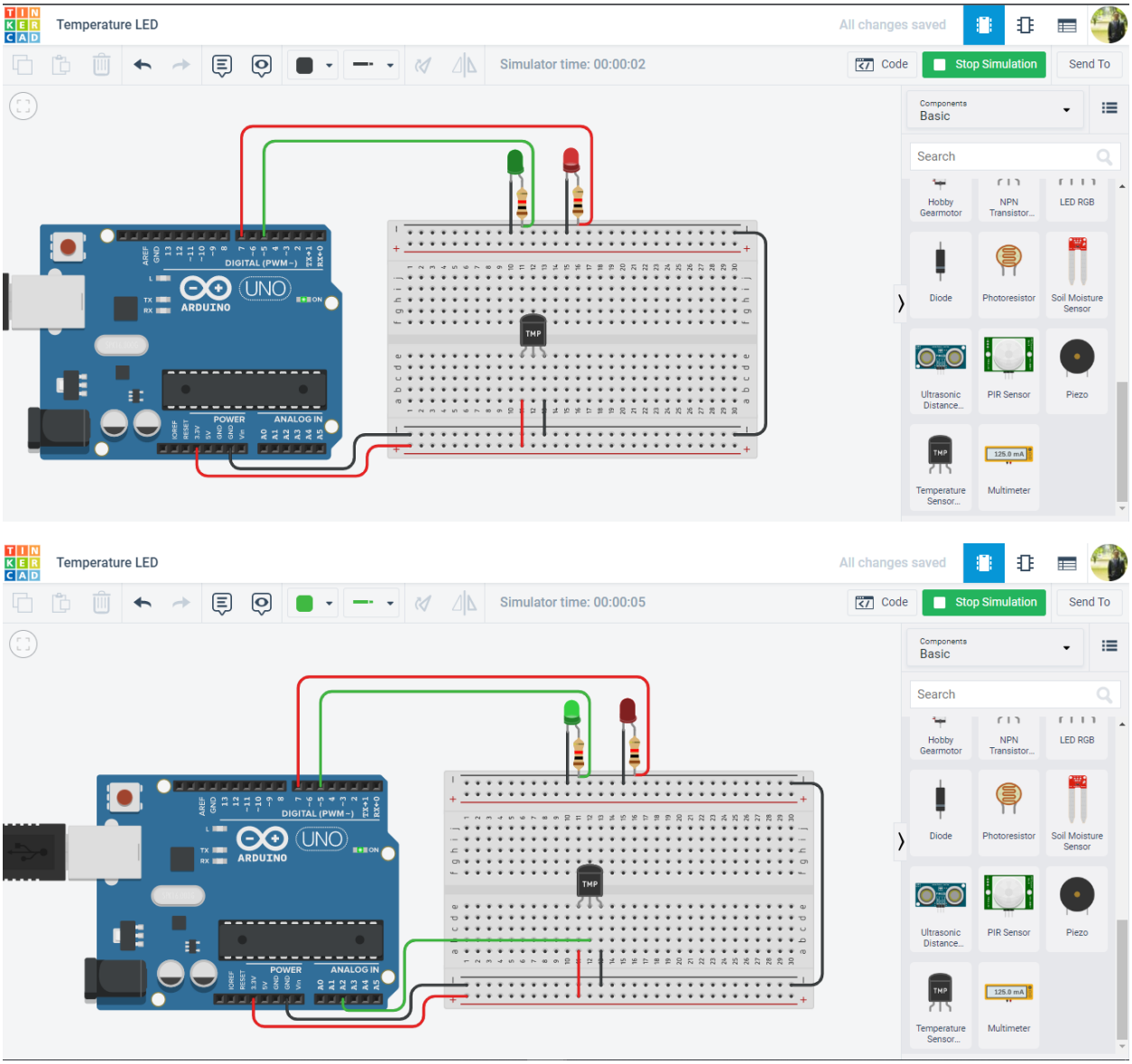

3 led sequence
All changes saved
Code
Start Simulation
Send To
Components
Basic
Search
Resistor
LED
Pushbutton
Potentiometer
Capacitor
Slideswitch
9V Battery
Coin Cell 3V Battery
1.5V Battery


3 led sequence
All changes saved
Simulator time: 00:00:12
Code
Stop Simulation
Send To
Components
Basic
Search
Resistor
LED
Pushbutton
Potentiometer
Capacitor
Slideswitch
9V Battery
Coin Cell 3V Battery
1.5V Battery

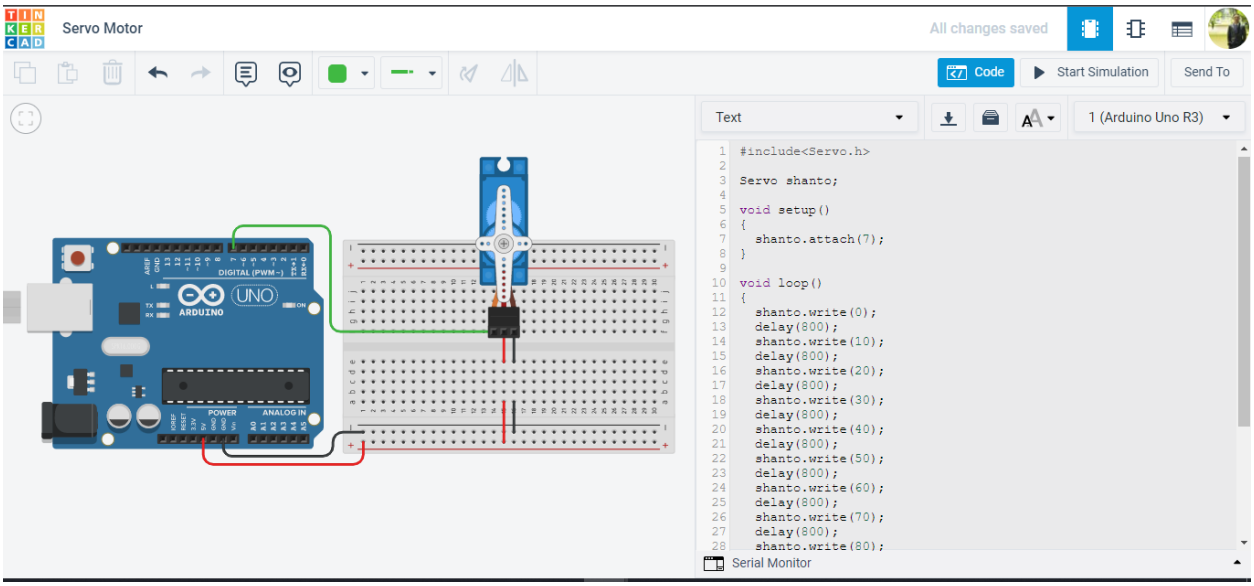


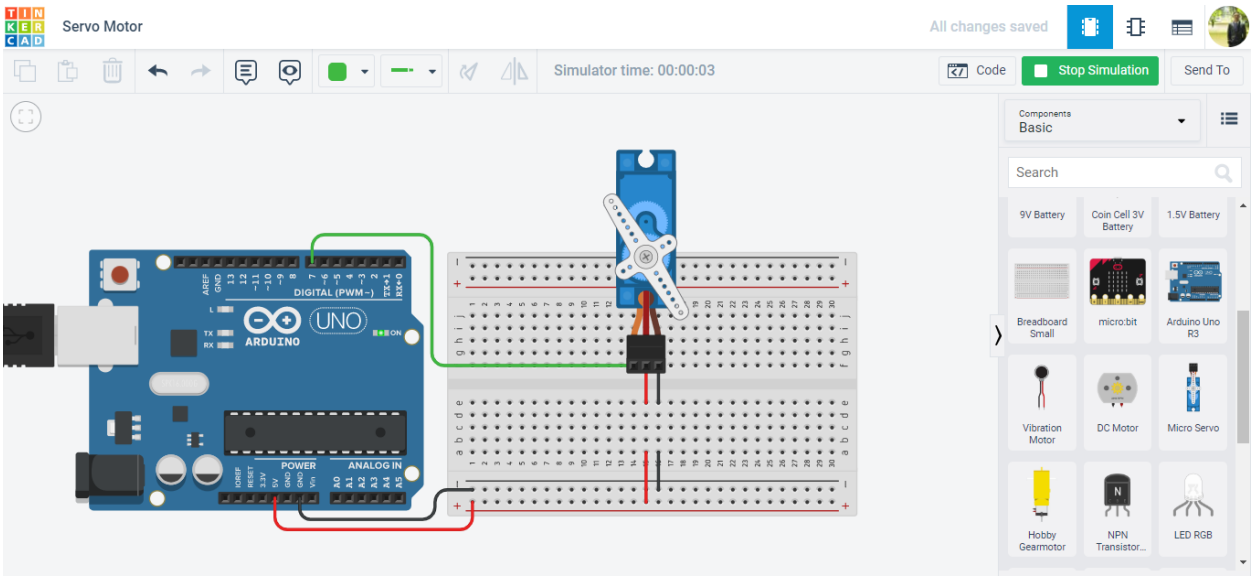
Turning LED on/off by TMP sensor

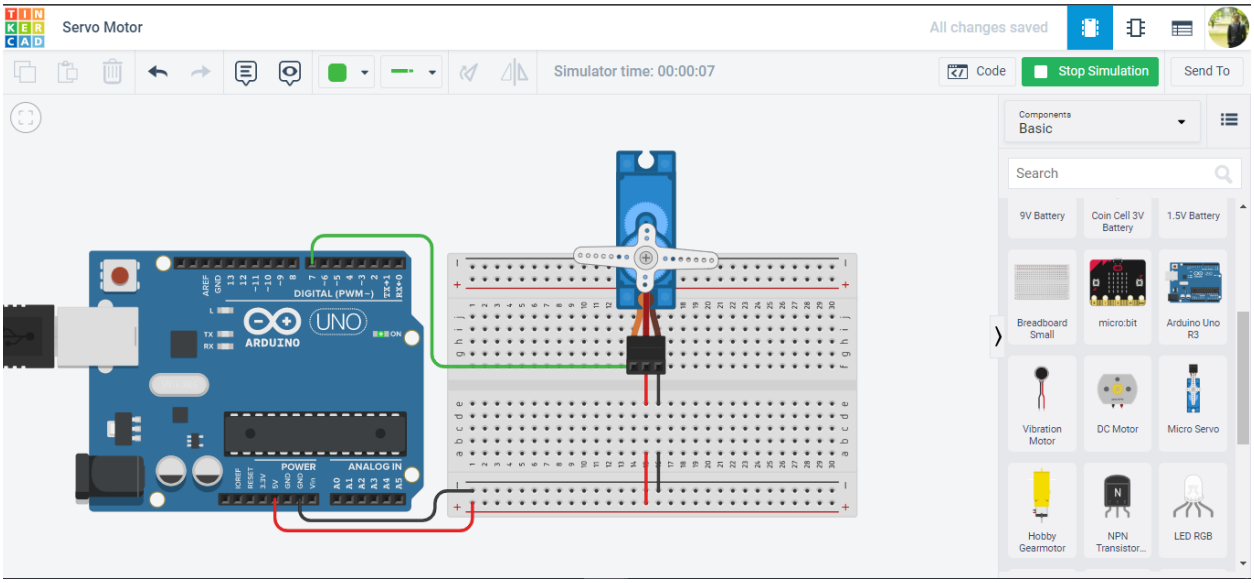




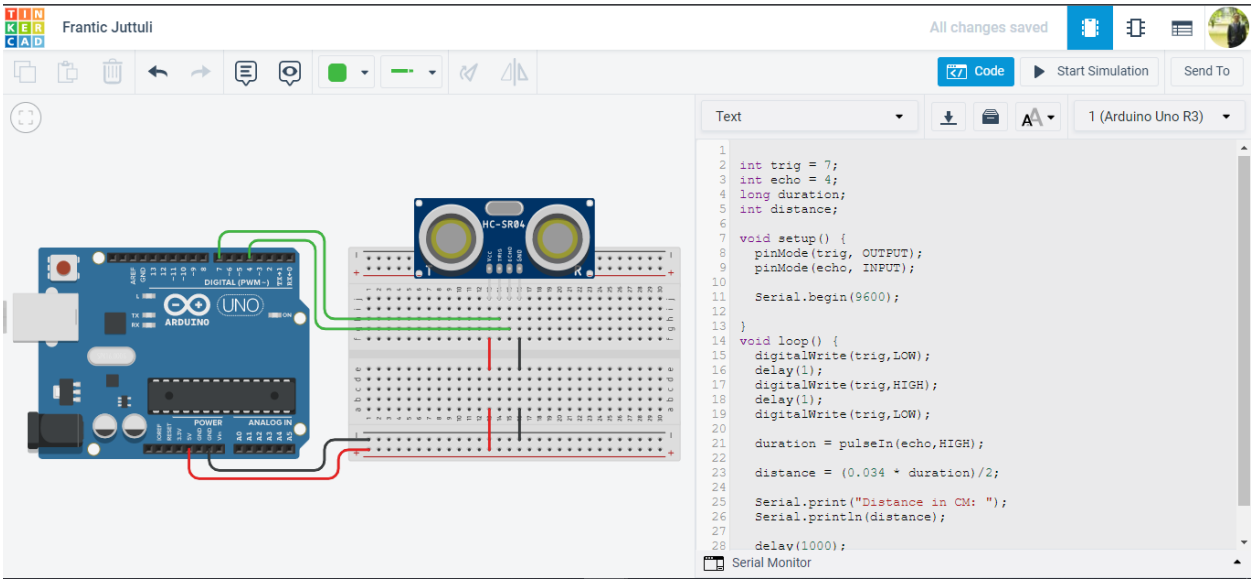
Rotation with servo motor

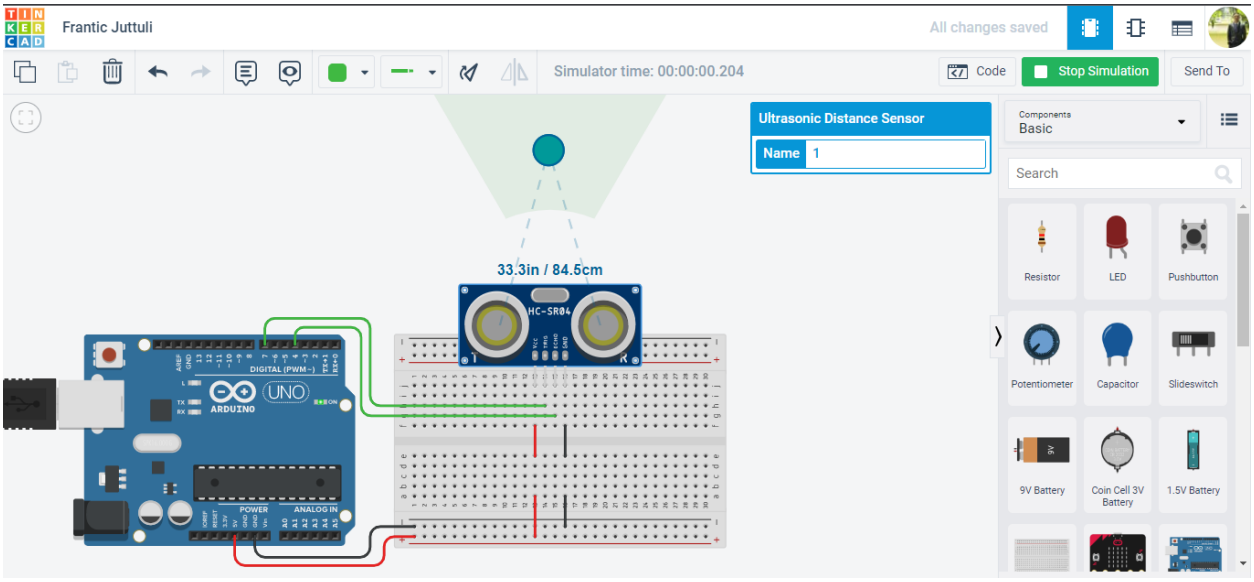

Servo Motor
All changes saved
Code
Start Simulation
Send To
Text
1 (Arduino Uno R3)
#include<Servo.h>
Servo shanto;
void setup()
{
shanto.attach(7);
}
void loop()
{
shanto.write(0);
delay(800);
shanto.write(10);
delay(800);
shanto.write(20);
delay(800);
shanto.write(30);
delay(800);
shanto.write(40);
delay(800);
shanto.write(50);
delay(800);
shanto.write(60);
delay(800);
shanto.write(70);
delay(800);
shanto.write(80);
Serial Monitor


Servo Motor
All changes saved
Simulator time: 00:00:03
Code
Stop Simulation
Send To
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




Frantic Juttuli
All changes saved
Simulator time: 00:00:00.204
Code
Stop Simulation
Send To
Ultrasonic Distance Sensor
Name
1
Components
Basic
Search
33.3in / 84.5cm
HC-SR04
Resistor
LED
Pushbutton
Potentiometer
Capacitor
Slideswitch
9V Battery
Coin Cell 3V Battery
1.5V Battery