الاسم: أحمد مجدى على حسين خليفة

الرقم الجامعى: 18010189

الاسم: محمد إبراهيم محمد السعيد إبراهيم

الرقم الجامعى: 18011342

الاسم: زياد السيد رمضان

الرقم الجامعى :18010716

الاسم: عبدالرحمن نبيل عبده محمد

الرقم الجامعى:18010991

الاسم: محمد ابراهيم جاد سعد

الرقم الجامعى:18011334

Microprocessor Software Project

Project #8: First-In-First-Out Queueing:

Introduction:

This program is taking an input (conditional input) to decide if someone entered the bank (input =1) so it will give him a number of the next available seat which he should set on or if someone left the bank (input =0) so it will detect that there is now an empty seat to count the available seat and organize the operation of seating in the bank.

Code:

```
#make_bin#


; BIN is plain binary format similar to .com format, but not limited to 1 segment;

; All values between # are directives, these values are saved into a separate .binf file.

; Before loading .bin file emulator reads .binf file with the same file name.



; All directives are optional, if you don't need them, delete them.


; set loading address, .bin file will be loaded to this address:

#LOAD_SEGMENT=0500h#

#LOAD_OFFSET=0000h#


; set entry point:

#CS=0500h#      ; same as loading segment

#IP=0000h#      ; same as loading offset


; set segment registers

#DS=0500h#      ; same as loading segment

#ES=0500h#      ; same as loading segment
```

```
; set stack
#SS=0500h#      ; same as loading segment
#SP=FFFEh#      ; set to top of loading segment
; set general registers (optional)
#AX=0000h#
#BX=0000h#
#CX=0000h#
#DX=0000h#
#SI=0000h#
#DI=0000h#
#BP=0000h#
include 'emu8086.inc'
MOV BX,0000h
MOV SI,0000h
MOV DI,0000h
MOV CX,10
EMPTY:
MOV [BX],0
INC BX
LOOP EMPTY
MOV BX,0000
PRINTN "ENTER 1 IF SOMEONE ENTERED THE BANK AND 0 IF SOMEONE LEFT "
PRINTN "############################################################"
INTRO:


PRINTN ""
```

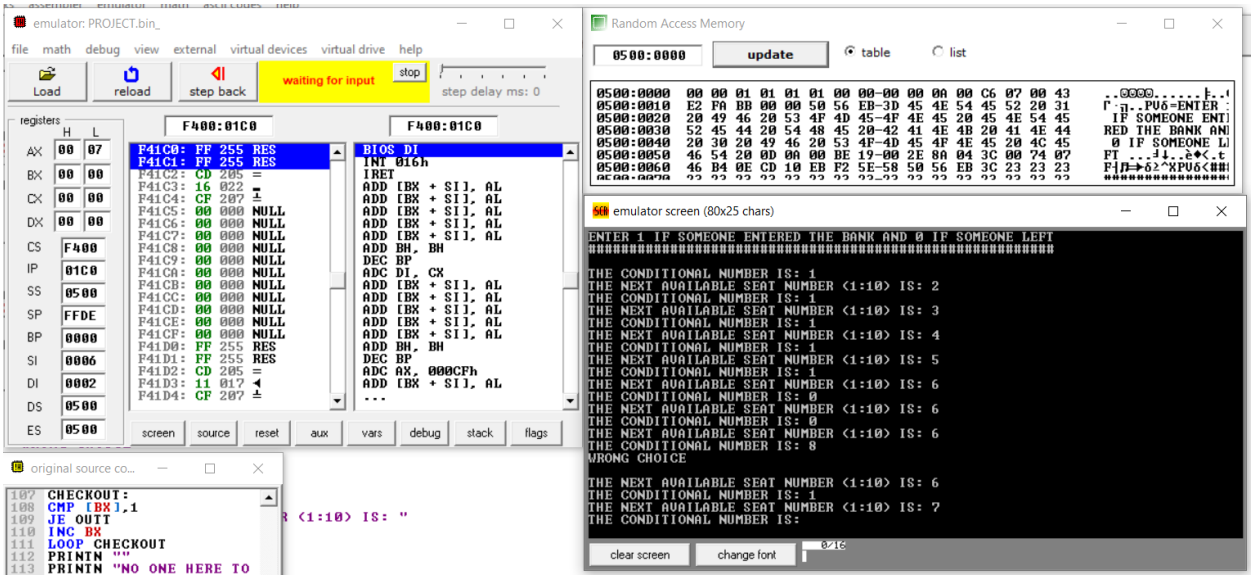
```
PRINT "THE CONDITIONAL NUMBER (0/1) IS: "

CALL SCAN_NUM

CMP CX,1
JE CHKINN

CMP CX,0
JE CHKOUT
JMP ERROR

OUTT:
MOV BX,0000
MOV [DI],0
INC DI
CMP DI,10
JE RETURNDI
JMP DETECT

ENTERED:
MOV BX,0000
MOV [SI],1
INC SI
CMP SI,10
JE RETURNSI
JMP DETECT
RETURNSI:
```

```
MOV SI,0
JMP DETECT
RETURNDI:
MOV DI,0
JMP DETECT
CHKINN:
MOV CX,10
CHECKIN:
CMP [BX],0
JE ENTERED
INC BX
LOOP CHECKIN
PRINTN ""
PRINTN "THERE IS NO EMPTY SEATS"
MOV BX,0
MOV SI,DI
JMP INTRO
CHKOUT:
MOV CX,10
CHECKOUT:
CMP [BX],1
JE OUTT
INC BX
LOOP CHECKOUT
PRINTN ""
PRINTN "NO ONE HERE TO LEAVE"
MOV DI,SI
```

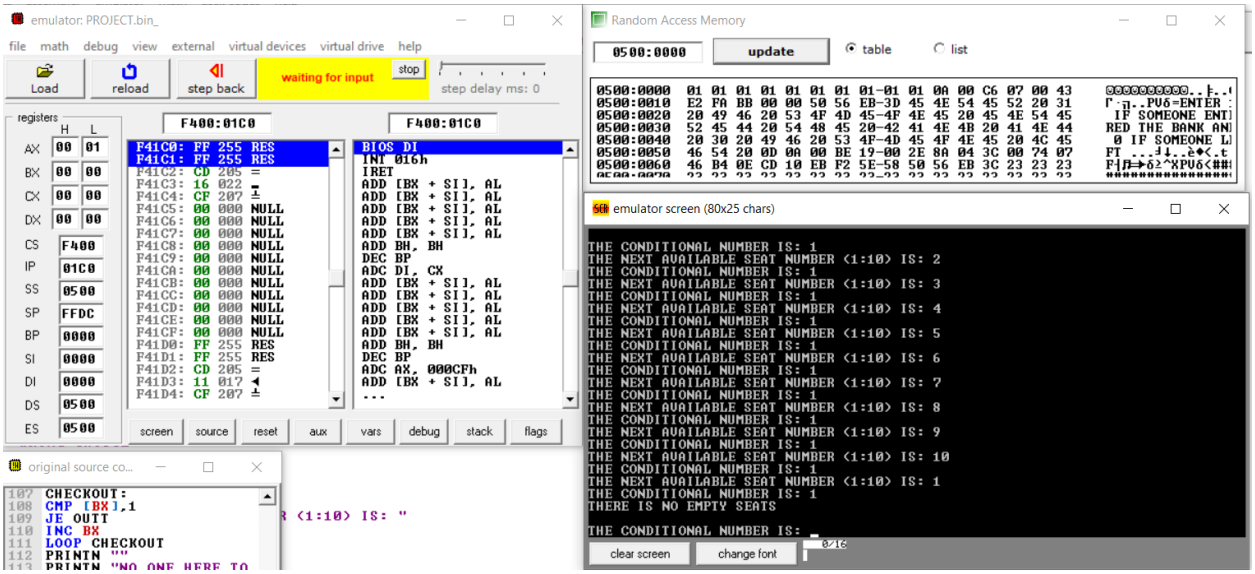
```
JMP INTRO

ERROR:

PRINTN ""

PRINTN "WRONG CHOICE"

DETECT:

MOV AX,SI

ADD AX,1

PRINTN ""

PRINT "THE NEXT AVAILABLE SEAT NUMBER (1:10) IS: "

CALL PRINT_NUM

CALL INTRO

DEFINE_SCAN_NUM

DEFINE_PRINT_STRING

DEFINE_PRINT_NUM

DEFINE_PRINT_NUM_UNS

DEFINE_PTHIS


HLT         ; halt!
```

Simulation:

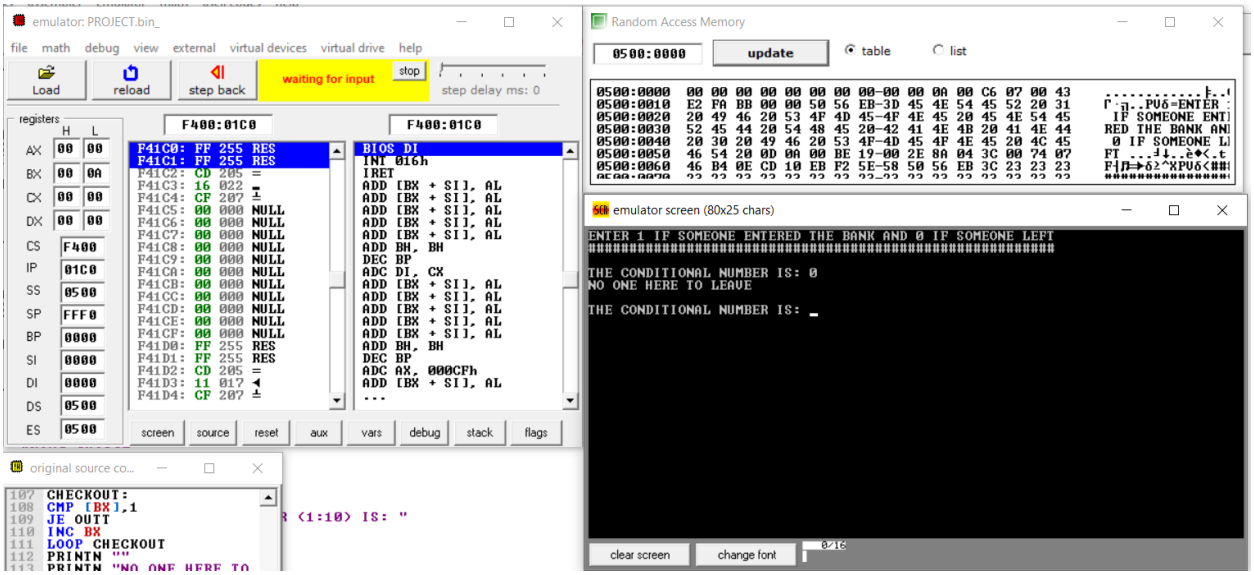
- Random experiment and if we tried to enter a wrong input.



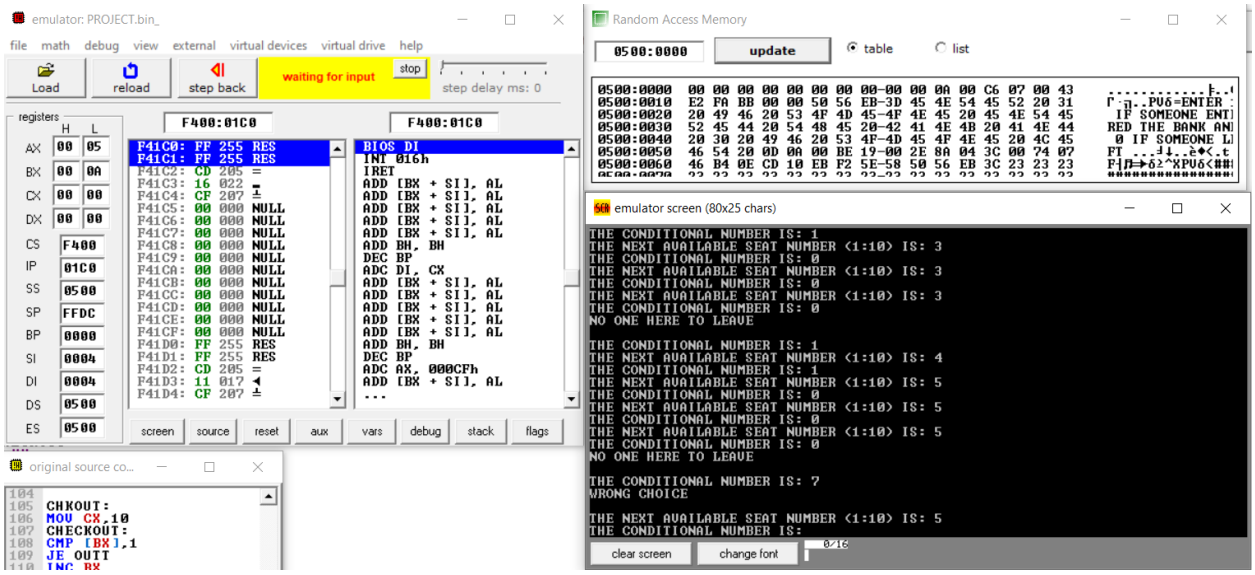
- If we entered 10 people and the seats are full.

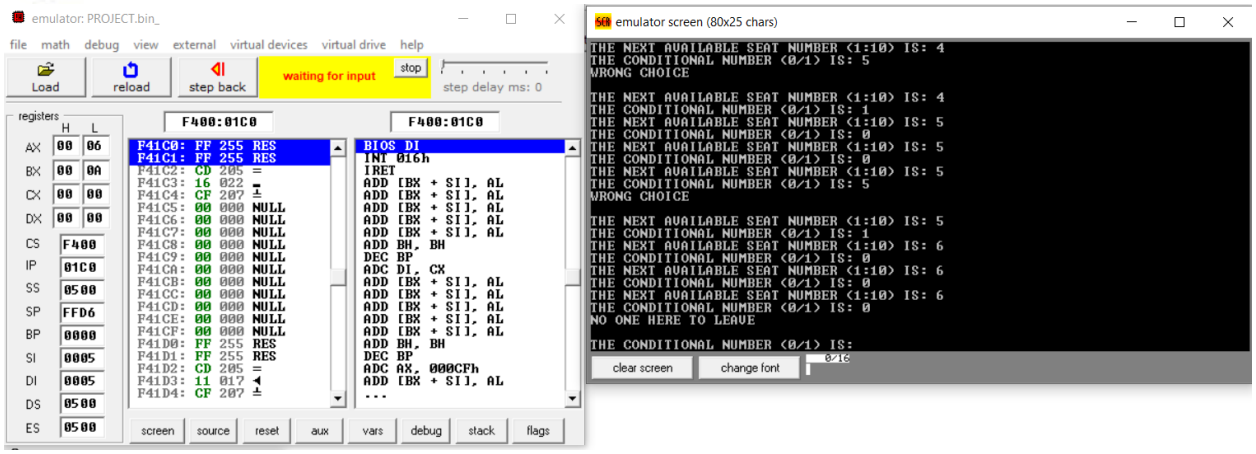


- If there is no one and we tried to get someone to leave.



- Random experiment.





All the simulations went perfect.