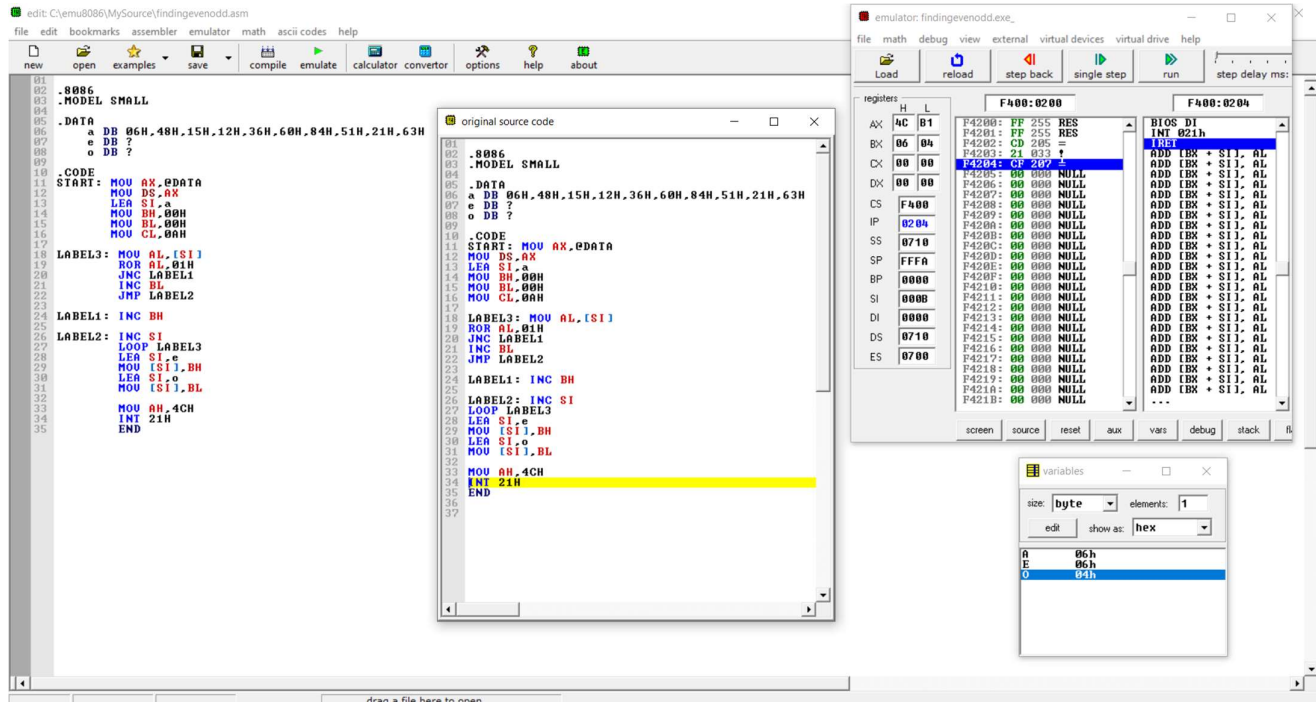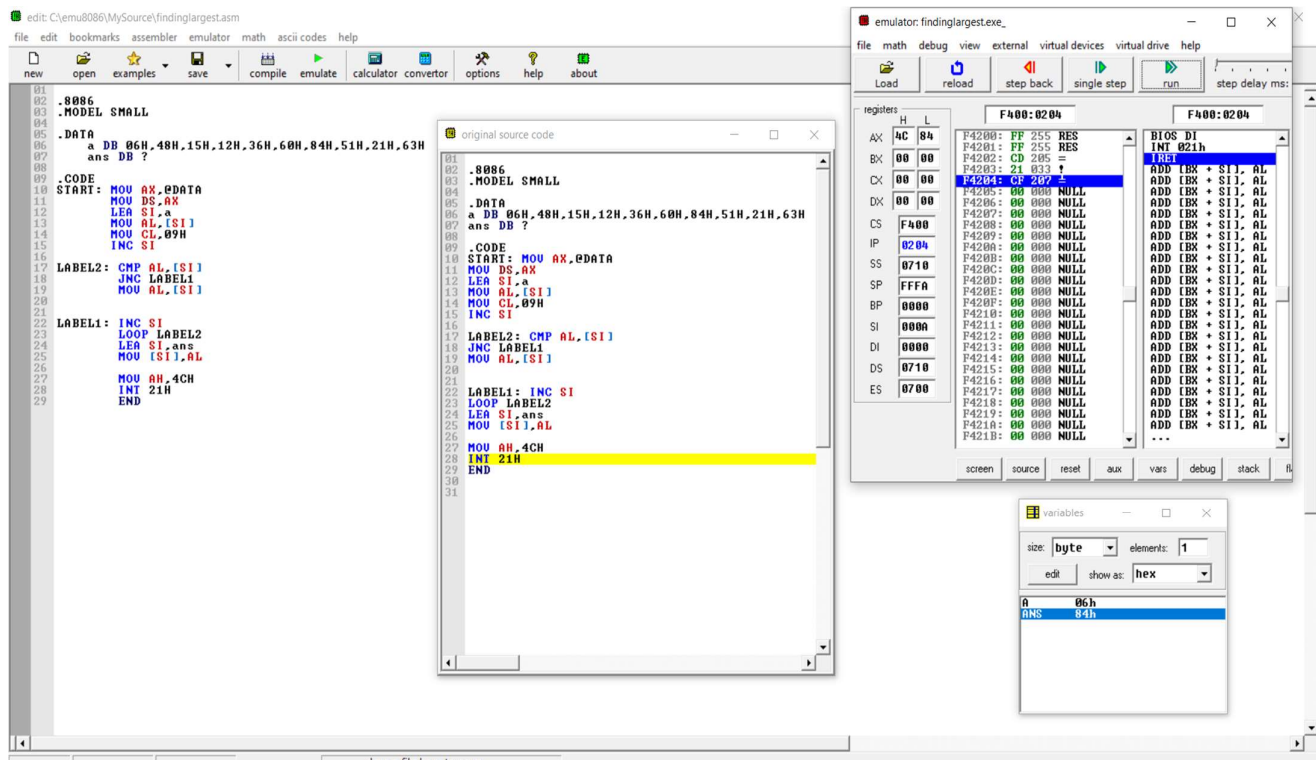Name: Brendan Lucas, Div: SE COMP B, Roll No: 8953

# Micro Processor Practical- 2.
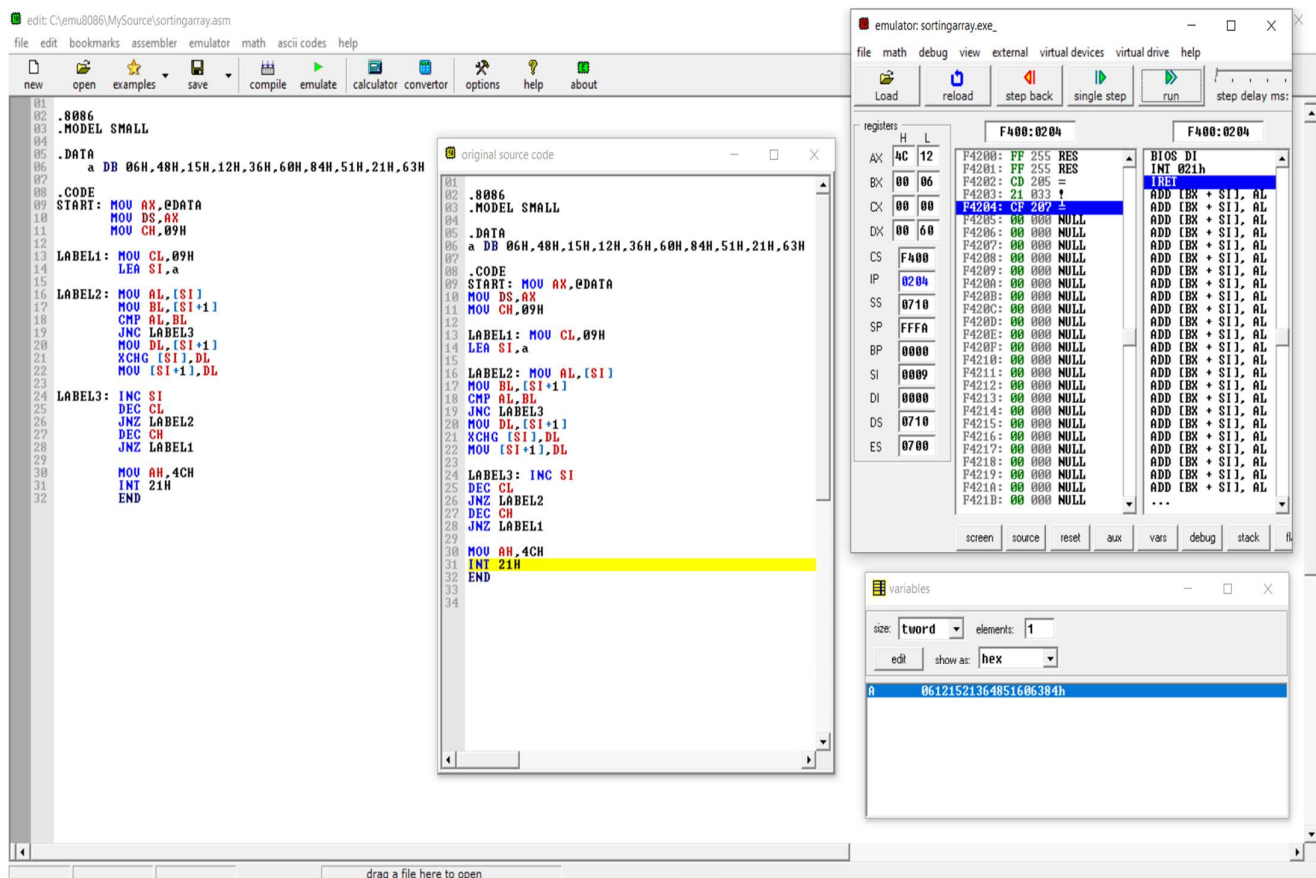
1. Counting Even and Odd numbers in an array:

## 2. Finding Largest number in an array of numbers:

edit: C:\emu8086\MySource\findinglargest.asm

file edit bookmarks assembler emulator math ascii codes help

new  open  examples  save  compile  emulate  calculator  convertor  options  help  about

```
.8086
.MODEL SMALL

.DATA
    a DB 06H,48H,15H,12H,36H,60H,84H,51H,21H,63H
    ans DB ?

.CODE
START: MOV AX,@DATA
       MOV DS,AX
       LEA SI,a
       MOV AL,[SI]
       MOV CL,09H
       INC SI

LABEL2: CMP AL,[SI]
        JNC LABEL1
        MOV AL,[SI]

LABEL1: INC SI
        LOOP LABEL2
        LEA SI,ans
        MOV [SI],AL

        MOV AH,4CH
        INT 21H
        END
```

emulator: findinglargest.exe_

file math debug view external virtual devices virtual drive help

Load  reload  step back  single step  run  step delay ms:

registers          F400:0204          F400:0204

|    | H | L |
|----|----|----|
| AX | 4C | 84 |
| BX | 00 | 00 |
| CX | 00 | 00 |
| DX | 00 | 00 |
| CS | F400 | |
| IP | 0204 | |
| SS | 0710 | |
| SP | FFFA | |
| BP | 0000 | |
| SI | 000A | |
| DI | 0000 | |
| DS | 0710 | |
| ES | 0700 | |

```
F4200: FF 255 RES
F4201: FF 255 RES
F4202: CD 205 =
F4203: 21 033 !
F4204: CF 207 ±
F4205: 00 000 NULL
F4206: 00 000 NULL
F4207: 00 000 NULL
F4208: 00 000 NULL
F4209: 00 000 NULL
F420A: 00 000 NULL
F420B: 00 000 NULL
F420C: 00 000 NULL
F420D: 00 000 NULL
F420E: 00 000 NULL
F420F: 00 000 NULL
F4210: 00 000 NULL
F4211: 00 000 NULL
F4212: 00 000 NULL
F4213: 00 000 NULL
F4214: 00 000 NULL
F4215: 00 000 NULL
F4216: 00 000 NULL
F4217: 00 000 NULL
F4218: 00 000 NULL
F4219: 00 000 NULL
F421A: 00 000 NULL
F421B: 00 000 NULL
```

```
BIOS DI
INT 021h
IRET
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
...
```

screen  source  reset  aux  vars  debug  stack  fl

original source code

```
.8086
.MODEL SMALL

.DATA
a DB 06H,48H,15H,12H,36H,60H,84H,51H,21H,63H
ans DB ?

.CODE
START: MOV AX,@DATA
       MOV DS,AX
       LEA SI,a
       MOV AL,[SI]
       MOV CL,09H
       INC SI

LABEL2: CMP AL,[SI]
        JNC LABEL1
        MOV AL,[SI]

LABEL1: INC SI
        LOOP LABEL2
        LEA SI,ans
        MOV [SI],AL

        MOV AH,4CH
        INT 21H
        END
```

variables

size: byte    elements: 1

edit    show as: hex

| A   | 06h |
|-----|-----|
| ANS | 84h |

drag a file here to open

## 3. Sorting an array of numbers in ascending order:

edit: C:\emu8086\MySource\sortingarray.asm

file edit bookmarks assembler emulator math ascii codes help

new  open  examples  save  compile  emulate  calculator  convertor  options  help  about

```
.8086
.MODEL SMALL

.DATA
    a DB 06H,48H,15H,12H,36H,60H,84H,51H,21H,63H

.CODE
START: MOV AX,@DATA
       MOV DS,AX
       MOV CH,09H

LABEL1: MOV CL,09H
        LEA SI,a

LABEL2: MOV AL,[SI]
        MOV BL,[SI+1]
        CMP AL,BL
        JNC LABEL3
        MOV DL,[SI+1]
        XCHG [SI],DL
        MOV [SI+1],DL

LABEL3: INC SI
        DEC CL
        JNZ LABEL2
        DEC CH
        JNZ LABEL1

        MOV AH,4CH
        INT 21H
        END
```

emulator: sortingarray.exe_

file math debug view external virtual devices virtual drive help

Load  reload  step back  single step  run  step delay ms:

registers          F400:0204          F400:0204

|    | H | L |
|----|----|----|
| AX | 4C | 12 |
| BX | 00 | 06 |
| CX | 00 | 00 |
| DX | 00 | 60 |
| CS | F400 | |
| IP | 0204 | |
| SS | 0710 | |
| SP | FFFA | |
| BP | 0000 | |
| SI | 0009 | |
| DI | 0000 | |
| DS | 0710 | |
| ES | 0700 | |

```
F4200: FF 255 RES
F4201: FF 255 RES
F4202: CD 205 =
F4203: 21 033 !
F4204: CF 207 ±
F4205: 00 000 NULL
F4206: 00 000 NULL
F4207: 00 000 NULL
F4208: 00 000 NULL
F4209: 00 000 NULL
F420A: 00 000 NULL
F420B: 00 000 NULL
F420C: 00 000 NULL
F420D: 00 000 NULL
F420E: 00 000 NULL
F420F: 00 000 NULL
F4210: 00 000 NULL
F4211: 00 000 NULL
F4212: 00 000 NULL
F4213: 00 000 NULL
F4214: 00 000 NULL
F4215: 00 000 NULL
F4216: 00 000 NULL
F4217: 00 000 NULL
F4218: 00 000 NULL
F4219: 00 000 NULL
F421A: 00 000 NULL
F421B: 00 000 NULL
```

```
BIOS DI
INT 021h
IRET
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
...
```

screen  source  reset  aux  vars  debug  stack  fl

original source code

```
.8086
.MODEL SMALL

.DATA
a DB 06H,48H,15H,12H,36H,60H,84H,51H,21H,63H

.CODE
START: MOV AX,@DATA
       MOV DS,AX
       MOV CH,09H

LABEL1: MOV CL,09H
        LEA SI,a

LABEL2: MOV AL,[SI]
        MOV BL,[SI+1]
        CMP AL,BL
        JNC LABEL3
        MOV DL,[SI+1]
        XCHG [SI],DL
        MOV [SI+1],DL

LABEL3: INC SI
        DEC CL
        JNZ LABEL2
        DEC CH
        JNZ LABEL1

        MOV AH,4CH
        INT 21H
        END
```

variables

size: tword    elements: 1

edit    show as: hex

| A | 06121521364851606384h |
|---|------------------------|

drag a file here to open

## 4. Unpacking a Packed BCD number:



```asm
.8086
.MODEL SMALL

.DATA
    a DB 23H
    num1 DB ?
    num2 DB ?

.CODE
START:  MOV AX,@DATA
        MOV DS,AX
        LEA SI,a
        MOV AL,[SI]
        MOV CL,04H
        AND AL,0FH
        MOV BL,AL
        MOV AL,[SI]
        SHR AL,CL
        MOV BH,AL
        LEA SI,num1
        MOV [SI],BH
        LEA SI,num2
        MOV [SI],BL
        MOV AH,4CH
        INT 21H
        END
```

## 5. Packing two Unpacked BCD numbers:



```asm
.8086
.MODEL SMALL

.DATA
    num1 DB 7H
    num2 DB 3H
    a DB ?

.CODE
START:  MOV AX,@DATA
        MOV DS,AX
        LEA SI,num1
        MOV AL,[SI]
        SHL AL,04H
        LEA SI,num2
        ADD AL,[SI]
        LEA SI,a
        MOV [SI],AL
        MOV AH,4CH
        INT 21H
        END
```