



Daffodil
International
University

Lab Report

Course Name: Microprocessor,
Embedded Systems & IoT lab
Course Code: CSE 232

Submitted By:

Mahadi Hasan Munna

ID: 203-15-3881
Section: PC – A
Department of CSE

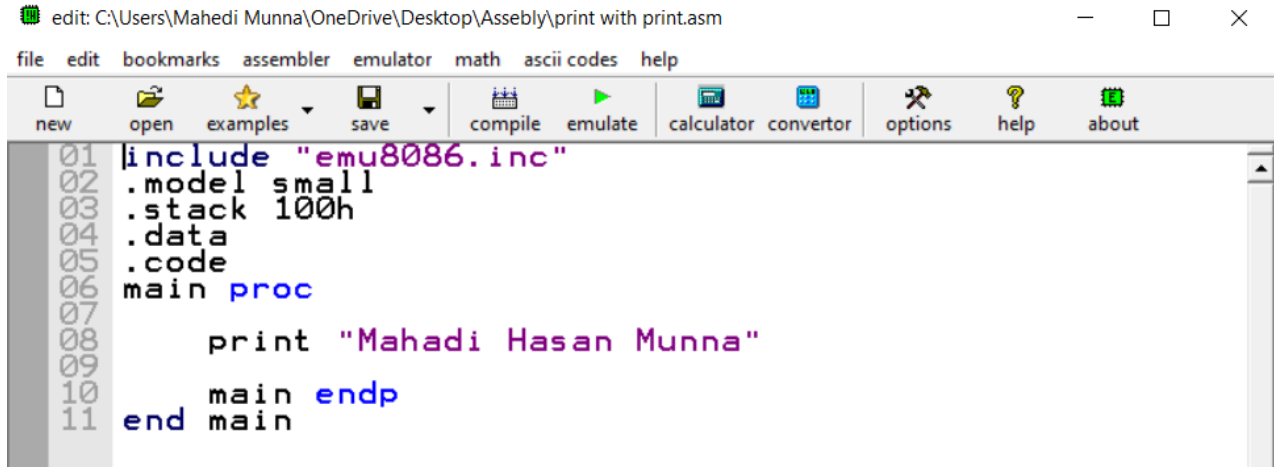
Submitted To:

Sonia Nasrin

Lecturer
Department of CSE
Daffodil International University

Lab 1

1. Printing my name with print function.



The screenshot shows an assembly editor window titled "edit: C:\Users\Mahedi Munna\OneDrive\Desktop\Asseby\print with print.asm". The menu bar includes file, edit, bookmarks, assembler, emulator, math, ascii codes, and help. The toolbar contains icons for new, open, examples, save, compile, emulate, calculator, convertor, options, help, and about. The code is as follows:

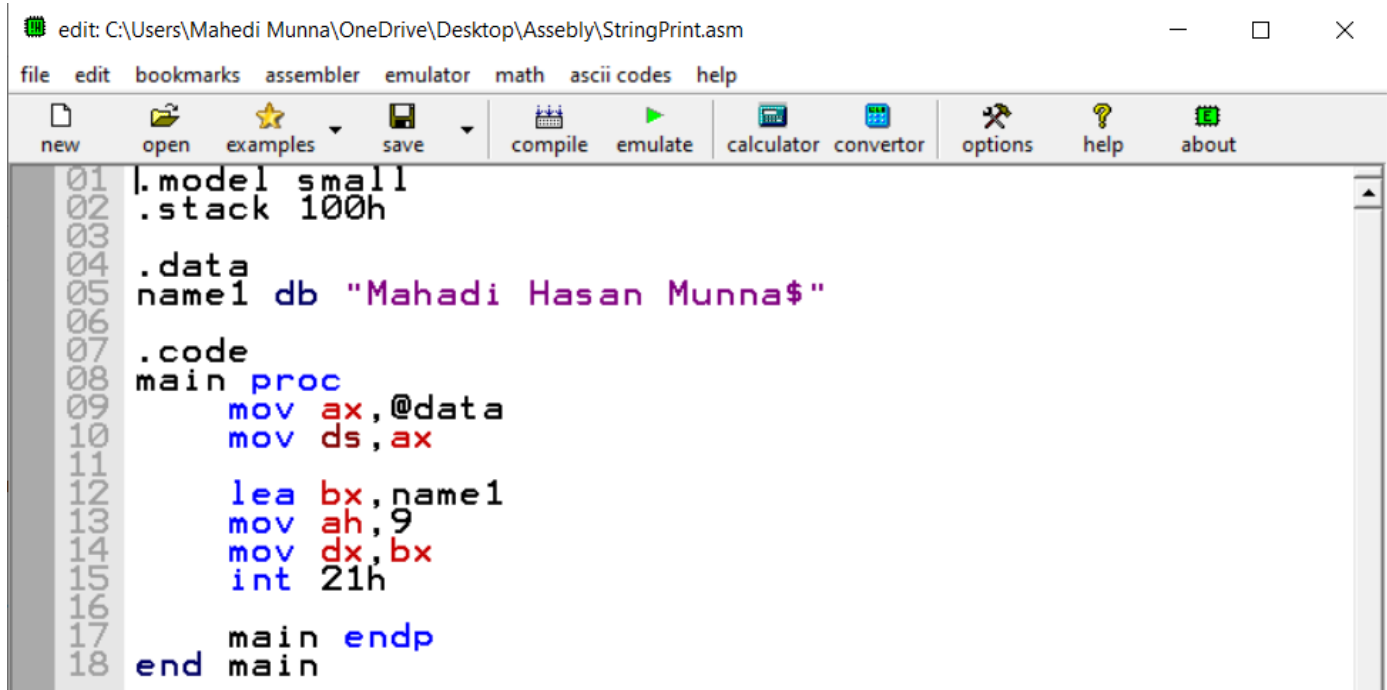
```
01 include "emu8086.inc"
02 .model small
03 .stack 100h
04 .data
05 .code
06 main proc
07
08     print "Mahadi Hasan Munna"
09
10     main endp
11 end main
```

Output:



The screenshot shows an emulator screen titled "emulator screen (80x25 chars)". The screen displays the output of the assembly program, which is the name "Mahadi Hasan Munna".

2. Printing my name with assembly



The screenshot shows an assembly editor window titled "edit: C:\Users\Mahedi Munna\OneDrive\Desktop\Asseby\StringPrint.asm". The menu bar includes file, edit, bookmarks, assembler, emulator, math, ascii codes, and help. The toolbar contains icons for new, open, examples, save, compile, emulate, calculator, convertor, options, help, and about. The code is as follows:

```
01 .model small
02 .stack 100h
03
04 .data
05 name1 db "Mahadi Hasan Munna$"
06
07 .code
08 main proc
09     mov ax,@data
10     mov ds,ax
11
12     lea bx,name1
13     mov ah,9
14     mov dx,bx
15     int 21h
16
17     main endp
18 end main
```

Output:

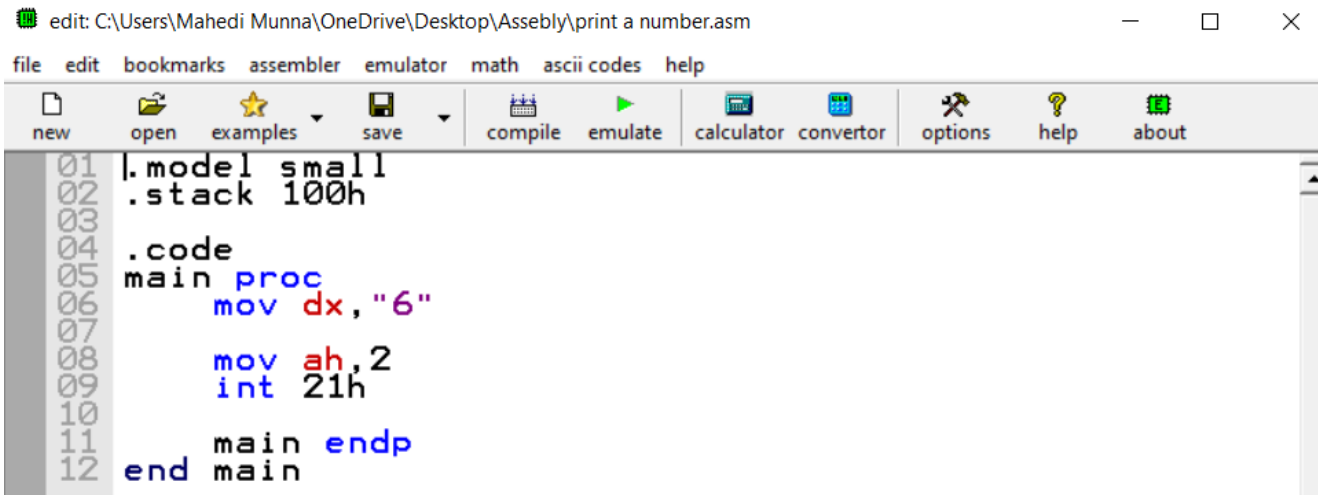


emulator screen (80x25 chars)

```
Mahadi Hasan Munna
```

Lab 2

1. Print a number



edit: C:\Users\Mahedi Munna\OneDrive\Desktop\Asseby\print a number.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 |
04 |.code
05 |main proc
06 |    mov dx,"6"
07 |
08 |    mov ah,2
09 |    int 21h
10 |
11 |    main endp
12 |end main
```

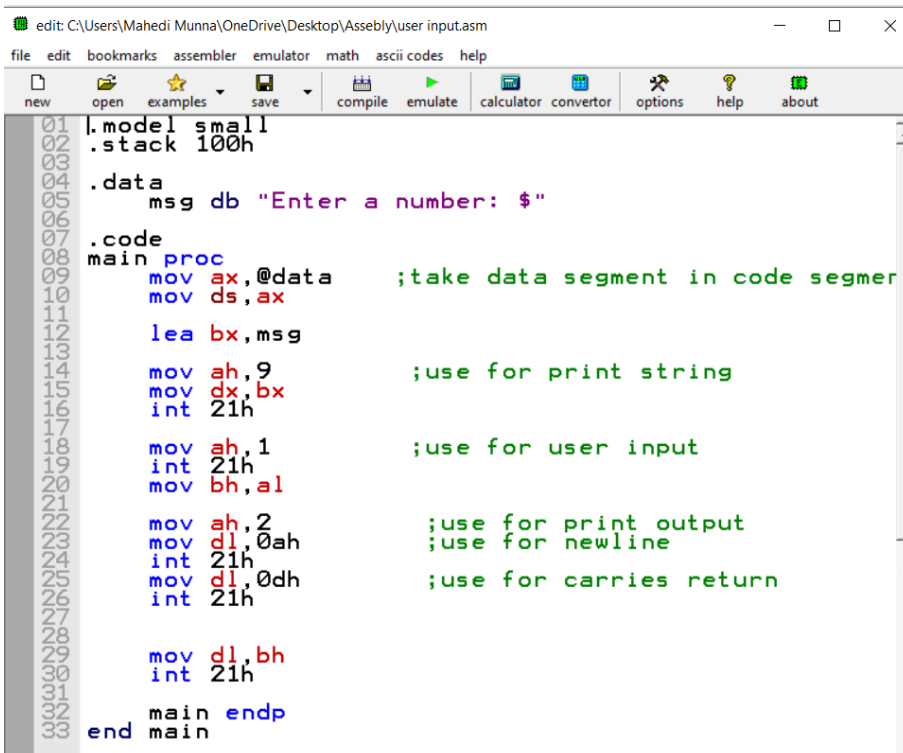
Output:



emulator screen (80x25 chars)

```
6
```

2. Take a user input and print it.



edit: C:\Users\Mahedi Munna\OneDrive\Desktop\Asseby\user input.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 |
04 |.data
05 |    msg db "Enter a number: $"
06 |
07 |.code
08 |main proc
09 |    mov ax,@data ;take data segment in code segmen
10 |    mov ds,ax
11 |
12 |    lea bx,msg
13 |
14 |    mov ah,9 ;use for print string
15 |    mov dx,bx
16 |    int 21h
17 |
18 |    mov ah,1 ;use for user input
19 |    int 21h
20 |    mov bh,al
21 |
22 |    mov ah,2 ;use for print output
23 |    mov dl,0ah ;use for newline
24 |    int 21h
25 |    mov dl,0dh ;use for carries return
26 |    int 21h
27 |
28 |
29 |    mov dl,bh
30 |    int 21h
31 |
32 |    main endp
33 |end main
```

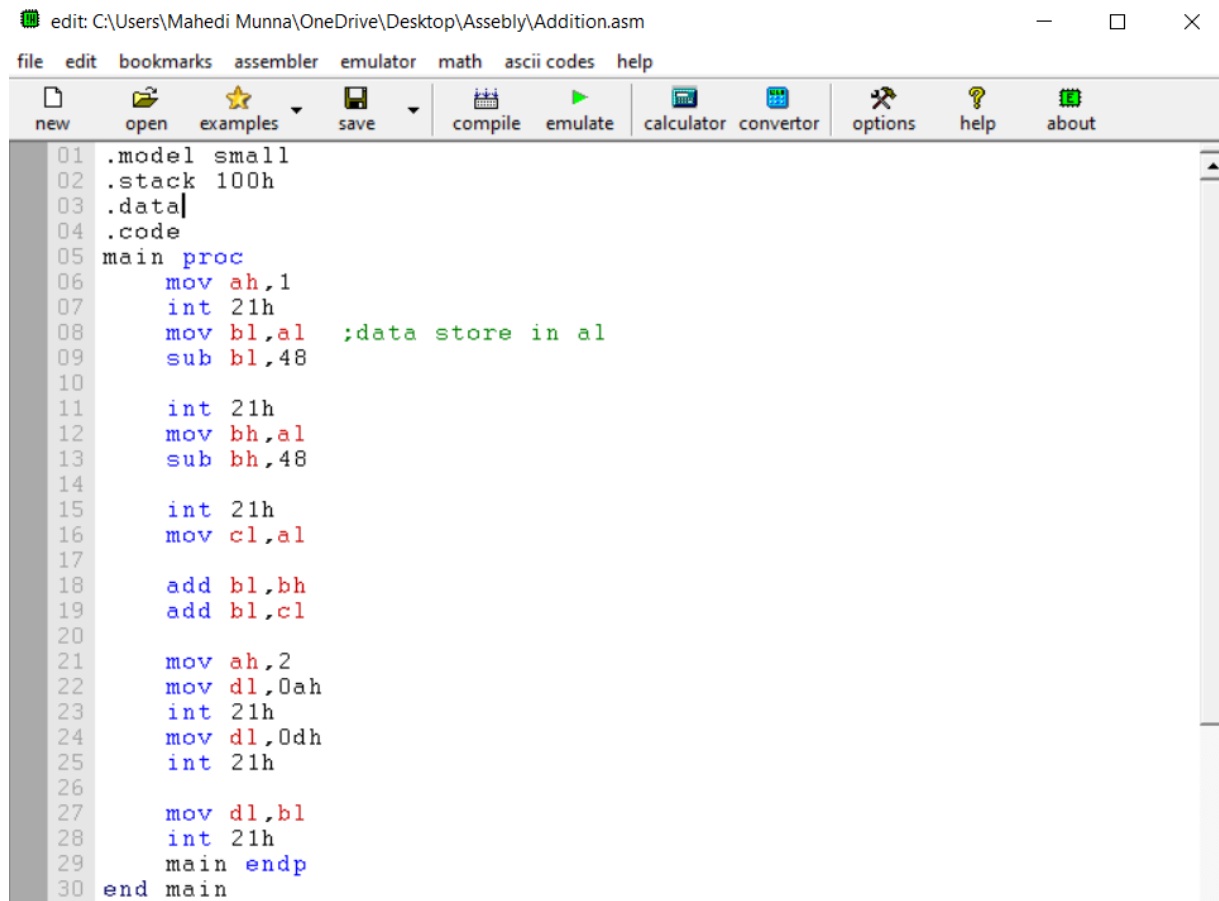
Output:



emulator screen (80x25 chars)

```
Enter a number: 5
5
```

3. Add 3 Numbers



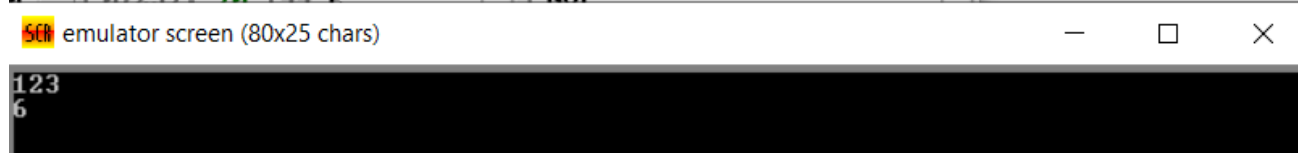
edit: C:\Users\Mahedi Munna\OneDrive\Desktop\Asseby\Addition.asm

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 main proc
06     mov ah,1
07     int 21h
08     mov bl,al    ;data store in al
09     sub bl,48
10
11     int 21h
12     mov bh,al
13     sub bh,48
14
15     int 21h
16     mov cl,al
17
18     add bl,bh
19     add bl,cl
20
21     mov ah,2
22     mov dl,0ah
23     int 21h
24     mov dl,0dh
25     int 21h
26
27     mov dl,bl
28     int 21h
29     main endp
30 end main
```

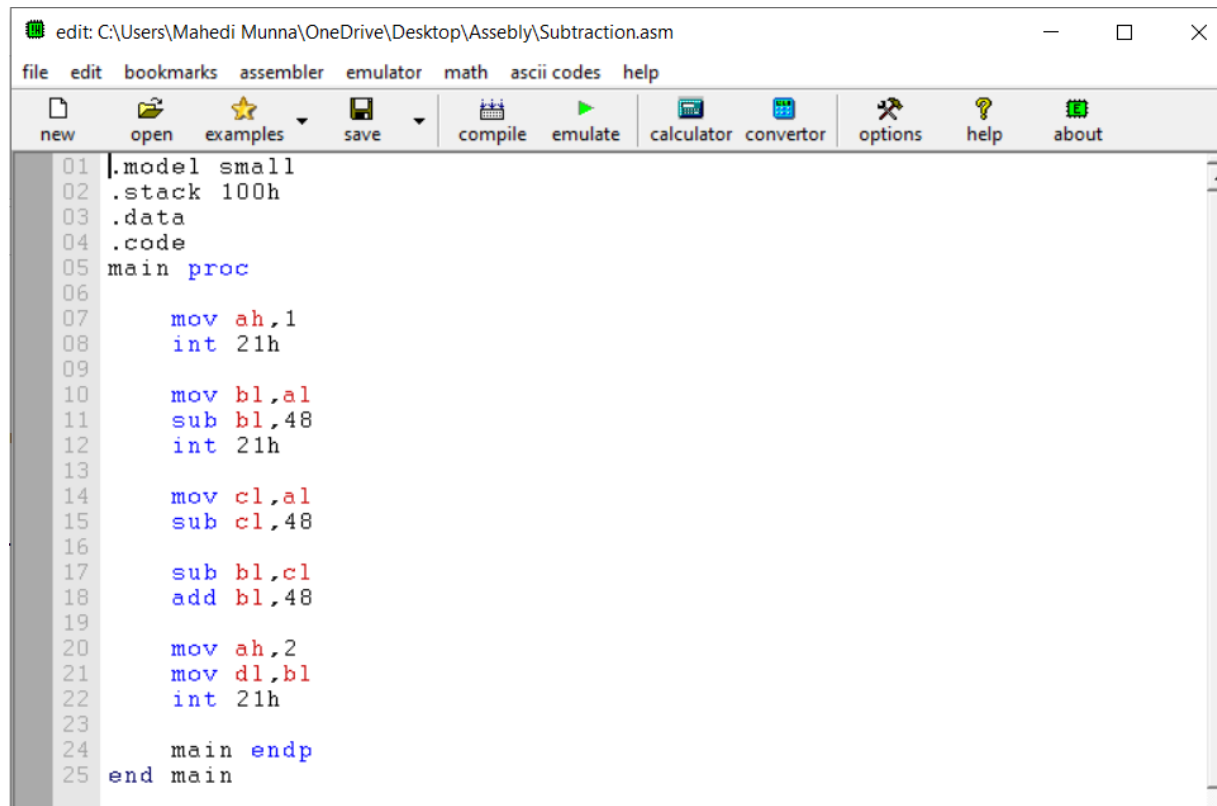
Output:



emulator screen (80x25 chars)

```
123
6
```

4. Subtraction 2 numbers



The screenshot shows the MASM6401 IDE with the file 'Subtraction.asm' open. The code is as follows:

```
01 .model small
02 .stack 100h
03 .data
04 .code
05 main proc
06
07     mov ah,1
08     int 21h
09
10     mov bl,a1
11     sub bl,48
12     int 21h
13
14     mov cl,a1
15     sub cl,48
16
17     sub bl,cl
18     add bl,48
19
20     mov ah,2
21     mov dl,bl
22     int 21h
23
24     main endp
25 end main
```

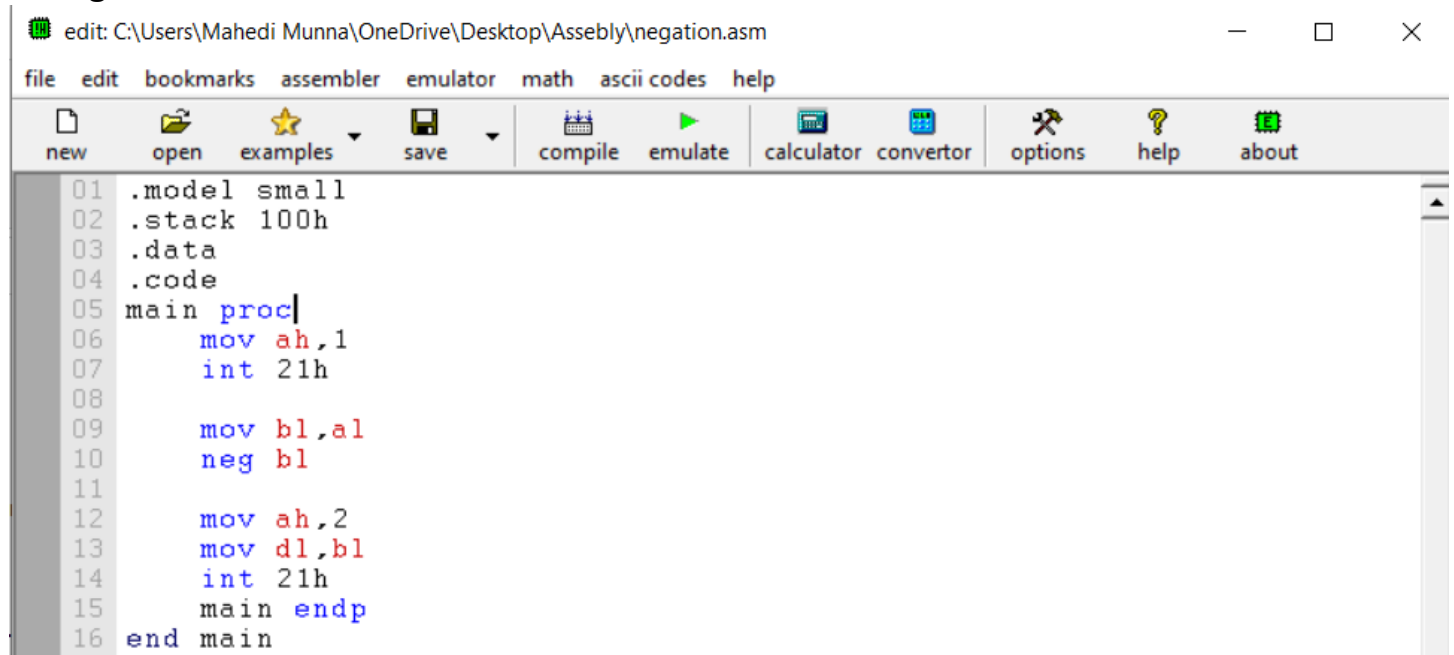
Output:



The screenshot shows the emulator screen with the title 'emulator screen (80x25 chars)'. The output displayed is:

```
633
```

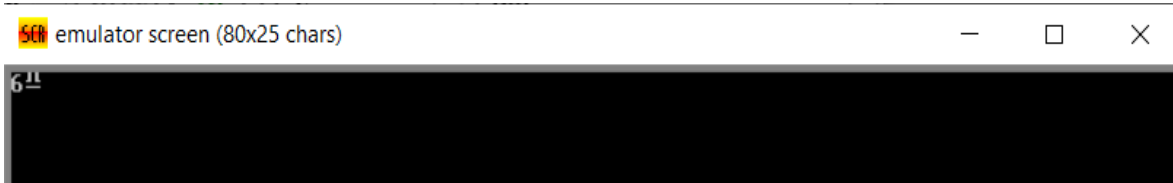
5. Negation



The screenshot shows the MASM6401 IDE with the file 'negation.asm' open. The code is as follows:

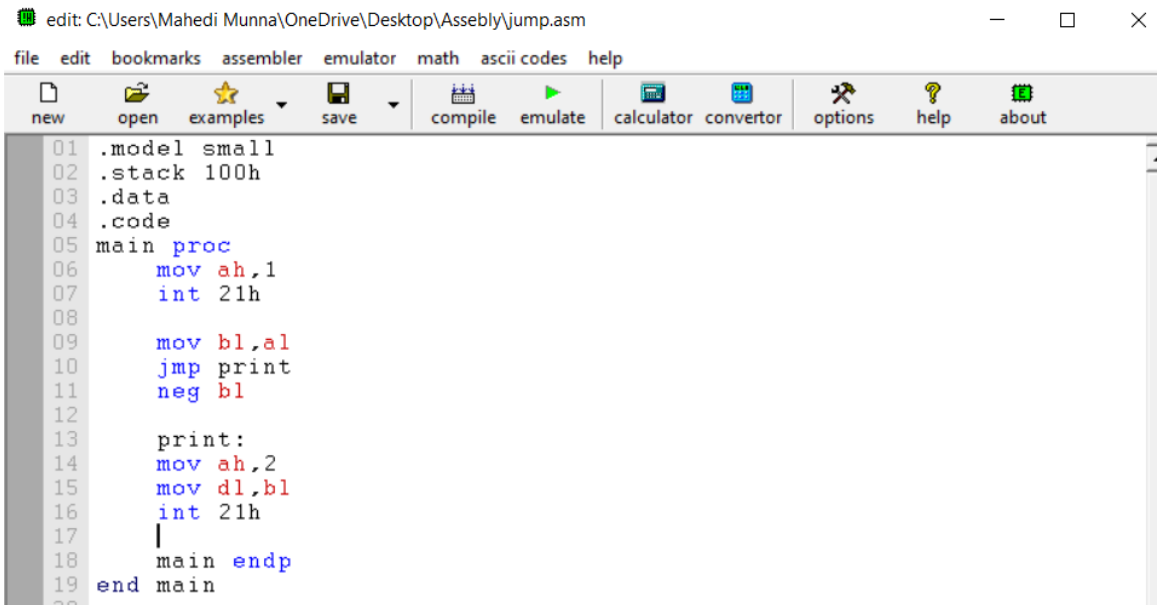
```
01 .model small
02 .stack 100h
03 .data
04 .code
05 main proc
06     mov ah,1
07     int 21h
08
09     mov bl,a1
10     neg bl
11
12     mov ah,2
13     mov dl,bl
14     int 21h
15     main endp
16 end main
```

Output:



Lab 3

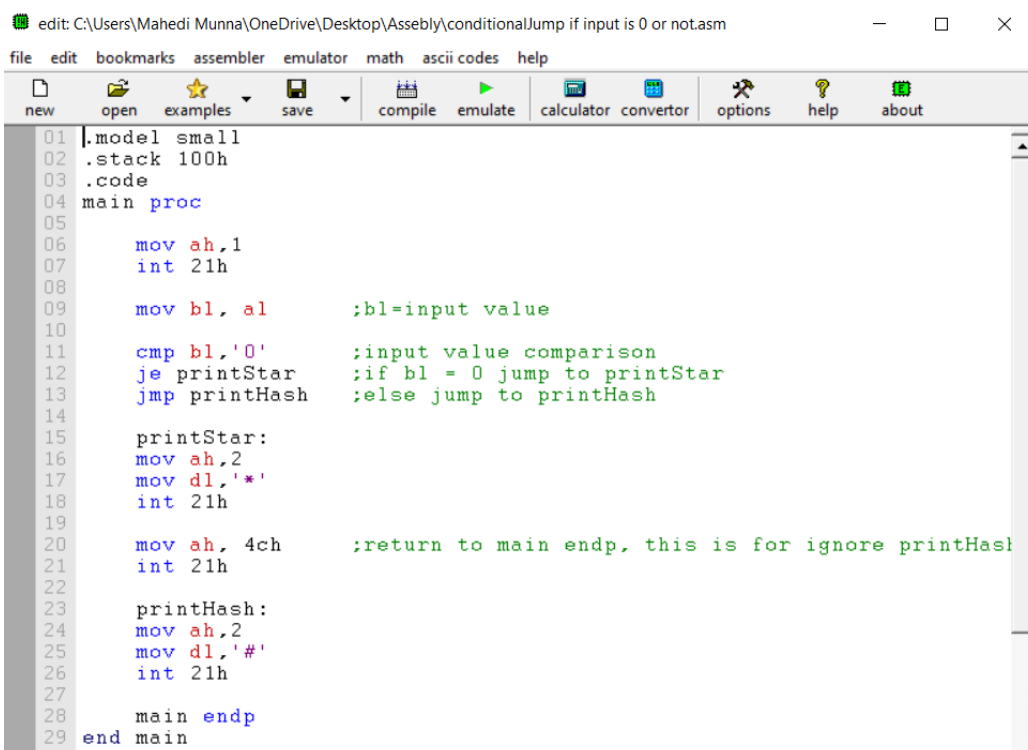
1.Jump



Output:



2. Conditional Jump [If zero then print * else #]



Output:



3. Conditional jump [Capital/small, vowel/not vowel, alphabet/not]

```
edit: C:\Users\Mahedi Munna\OneDrive\Desktop\Assembly\alpha
file edit bookmarks assembler emulator math ascii co
new open examples save compile err
001 .model small
002 .stack 100h
003 .data
004 msg1 db "Capital letter$"
005 msg2 db "Small letter$"
006 msg3 db "It is a vowel$"
007 msg4 db "It is not a vowel$"
008 msg5 db "Not an Alphabete$"
009
010 .code
011 main proc
012
013     mov ax,@data
014     mov ds,ax
015
016     mov ah,1
017     int 21h
018     mov bl,al
019
020     cmp bl,'a'
021     jl capital
022
023     cmp bl,'z'
024     jg capital
025
026     mov ah,2
027     mov dl,10
028     int 21h
029     mov dl,13
030     int 21h
031
032     lea cx,msg2
033     mov dx,cx
034     mov ah,9
035     int 21h
036
037     jmp vowel
038
039     capital:
040     cmp bl,'A'
041     jl else
042
043     cmp bl,'Z'
044     jg else
045
046     mov ah,2
047     mov dl,10
048     int 21h
049     mov dl,13
050     int 21h
051
052     lea cx,msg1
053     mov dx,cx
054     mov ah,9
055     int 21h
056
057     vowel:
058     cmp bl,'a'
059     je printVowel
060
061     cmp bl,'e'
062     je printVowel
063
064     cmp bl,'i'
065     je printVowel
066
067     cmp bl,'o'
068     je printVowel
069
070     cmp bl,'u'
071     je printVowel
072
073     cmp bl,'A'
074     je printVowel
075
076     cmp bl,'E'
077     je printVowel
078
079     cmp bl,'I'
080     je printVowel
081
082     cmp bl,'O'
083     je printVowel
084
085     cmp bl,'U'
086     je printVowel
087
088     jmp notVowel
089
090     printVowel:
091     mov ah,2
092     mov dl,10
093     int 21h
094     mov dl,13
095     int 21h
096
097     lea cx,msg3
098     mov dx,cx
099     mov ah,9
100     int 21h
101     jmp exit
102
103     notVowel:
104     mov ah,2
105     mov dl,10
106     int 21h
107     mov dl,13
108     int 21h
109
110     lea cx,msg4
111     mov dx,cx
112     mov ah,9
113     int 21h
114     jmp exit
115
116     else:
117     mov ah,2
118     mov dl,10
119     int 21h
120     mov dl,13
121     int 21h
122
123     lea cx,msg5
124     mov dx,cx
125     mov ah,9
126     int 21h
127
128     exit:
129     mov ah,4ch
130     int 21h
131
132     main endp
133 end main
134
```

Output:

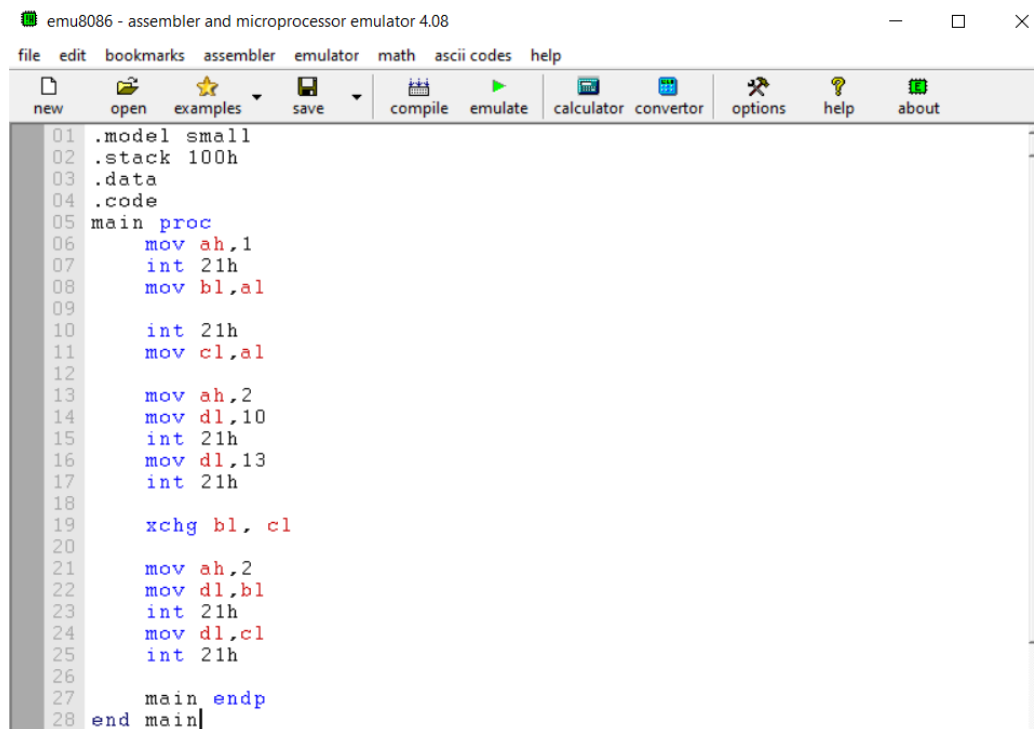


```
g
Small letter
It is not a vowel
```



```
O
Capital letter
It is a vowel
```

4. Exchange Number



```
01 .model small
02 .stack 100h
03 .data
04 .code
05 main proc
06     mov ah,1
07     int 21h
08     mov bl,al
09
10     int 21h
11     mov cl,al
12
13     mov ah,2
14     mov dl,10
15     int 21h
16     mov dl,13
17     int 21h
18
19     xchg bl, cl
20
21     mov ah,2
22     mov dl,bl
23     int 21h
24     mov dl,cl
25     int 21h
26
27     main endp
28 end main
```

Output:



```
45
54
```


Lab 4

1. Compare and print biggest

edit: C:\Users\Mahedi Munna\OneDrive\Desktop\Asseby\coparison bewteen two input.asm

file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator convertor

```
01 .model small
02 .stack 100h
03 .data
04     msg1 db "take two input $"
05     msg2 db "biggest value is: $"
06 .code
07 main proc
08
09 mov ax,@data
10 mov ds,ax
11
12 mov ah,9
13 lea dx,msg1
14 int 21h
15
16 mov ah,1
17 int 21h
18 mov bx,ax
19 int 21h
20
21 mov cx,ax
22
23 mov ah,2
24 mov dl,0ah
25 int 21h
26 mov dl,0dh
27 int 21h
28
29
30 cmp bx,cx    ; comparrison between two input
31 jg level1
32
33 jmp level2
34
35 level1:
36 mov ah,9
37 lea dx,msg2
38 int 21h
39
40 mov ah,2
41 mov dx,bx
42 int 21h
43
44 jmp exit
45
46 level2:
47 mov ah,9
48 lea dx,msg2
49 int 21h
50
51 mov ah,2
52 mov dx,cx
53 int 21h
54
55
56 exit:
57 mov ah,4ch
58 int 21h
59
60
61     main endp
62 end main
```

edit: C:\Users\Mahedi Munna\OneDrive\D

file edit bookmarks assembler emula

new open examples save

```
34
35 level1:
36 mov ah,9
37 lea dx,msg2
38 int 21h
39
40 mov ah,2
41 mov dx,bx
42 int 21h
43
44 jmp exit
45
46 level2:
47 mov ah,9
48 lea dx,msg2
49 int 21h
50
51 mov ah,2
52 mov dx,cx
53 int 21h
54
55
56 exit:
57 mov ah,4ch
58 int 21h
59
60
61     main endp
62 end main
```

drag a file here to

Output:

emulator screen (80x25 chars)

```
take two input 64
biggest value is: 6
```

2. Even or Odd between 1 – 4

```
edit: C:\Users\Mahedi Munna\OneDrive\Desktop\Asseblly\con
file  edit  bookmarks  assembler  emulator  math  ascii cc
new  open  examples  save  compile  er

01  .model small
02  .stack 100h
03  .code
04  main proc
05
06      mov ah,1
07      int 21h
08
09
10      ;1 or 3
11      cmp al,'1'
12      je odd
13      cmp al,'3'
14      je odd
15
16
17      ;2 or 4
18      cmp al,'2'
19      je even
20      cmp al,'4'
21      je even
22
23      jmp exit
24
25      odd:
26      mov ah,2
27      mov dl,'o'
28      int 21h
29
30      jmp exit
31
32      even:
33      mov ah,2
34      mov dl,'e'
35      int 21h
36
37      exit:
38      mov ah,4ch
39      int 21h
40
```

Output:

emulator screen (80x25 chars)

2e

Lab 5

1. Print name 10 times with loop

Output:

```
edit: C:\Users\Mahedi Munna\OneDrive\Desktop\Asseby\my na
file edit bookmarks assembler emulator math ascii code
new open examples save compile emul

01 |.model small
02 |.stack 100h
03 |.data
04 |name1 db "Mahadi Hasan Munna$"
05 |.code
06 |main proc
07 |
08 |
09 |     mov cx,10
10 |
11 |     mov ax,@data
12 |     mov ds,ax
13 |
14 |
15 |
16 |     repeat:
17 |     lea bx,name1
18 |     mov ah,9
19 |     mov dx,bx
20 |     int 21h
21 |
22 |     mov ah,2
23 |     mov dl,10
24 |     int 21h
25 |     mov dl,13
26 |     int 21h
27 |     loop repeat
28 |
29 |     mov ah,4ch
30 |     int 21h
31 |
32 |     main endp
33 |end main
```

SCN emulator screen (80x25 chars)

```
Mahadi Hasan Munna
Mahadi Hasan Munna
Mahadi Hasan Munna
Mahadi Hasan Munna
Mahadi Hasan Munna
Mahadi Hasan Munna
Mahadi Hasan Munna
Mahadi Hasan Munna
Mahadi Hasan Munna
Mahadi Hasan Munna
```

2. Print A to Z

Output:

```
edit: C:\Users\Mahedi Munna\OneDrive\Desktop\Asseby\loc
file edit bookmarks assembler emulator math ascii co
new open examples save compile e

01 |.model small
02 |.stack 100h
03 |.data
04 |.code
05 |main proc
06 |
07 |
08 |     mov cx,26
09 |
10 |     mov ah,2
11 |     mov bl,65
12 |
13 |     repeat:
14 |     mov dl,bl
15 |     int 21h
16 |     inc bl
17 |
18 |     mov dl,10
19 |     int 21h
20 |     mov dl,13
21 |     int 21h
22 |     loop repeat
23 |
24 |     mov ah,4ch
25 |     int 21h
26 |
27 |     main endp
28 |end main
```

SCN emulator screen (80x25 chars)

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

3. Print star with decrement

edit: C:\Users\Mahedi Munna\OneDrive\Desktop\Asseby\print star with

```
file edit bookmarks assembler emulator math ascii codes hel
new open examples save compile emulate
01 .model small
02 .stack 100h
03 .data
04 .code
05 main proc
06     mov ah,1
07     int 21h
08
09     mov cl,al
10     mov bl,cl
11
12     mov ah,2
13     mov dl,10
14     int 21h
15     mov dl,13
16     int 21h
17
18     event:
19     mov dl,'*'
20     int 21h
21
22     cmp bl,'1'
23     je exit
24
25     cmp cl,'1'
26     je outerLoop
27
28     jmp innerLoop
29
30     outerLoop:
31     mov ah,2
32     mov dl,10
33     int 21h
34     mov dl,13
35     int 21h
36
37     mov cl,bl
38     dec bl
39
40     innerLoop:
41     loop event
42
43     exit:
44     mov ah,4ch
```

Output:

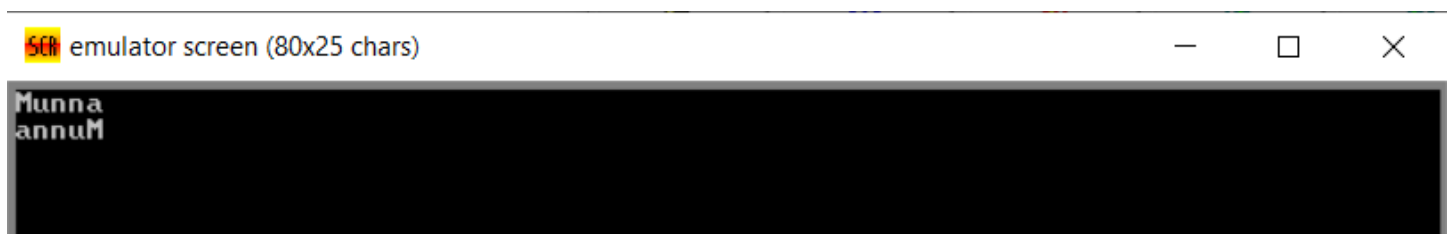
```
emulator screen (80x25 chars)
5
*****
****
***
**
*
```

Lab 6

1.Reverse a string

```
edit: C:\Users\Mahedi Munna\OneDrive\Desktop\Asseby\Stacking and reversing.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator conve
01 .model small
02 .stack 100h
03 .data
04 .code
05 main proc
06     xor cx, cx
07
08     mov ah, 1
09     int 21h
10
11     compare:
12     cmp al, 13
13     je output
14     inc cx
15     push ax
16     int 21h
17     jmp compare
18
19     output:
20     mov ah, 2
21     mov dl, 10
22     int 21h
23     mov dl, 13
24     int 21h
25
26     execute:
27     pop dx
28     int 21h
29     loop execute
30
31     mov ah, 2
32     mov dl, 10
33     int 21h
34     mov dl, 13
35     int 21h
36
37     mov ah, 2
38     mov dx, cx
39     int 21h
40
41     main endp
42 end main
```

Output:



2. Count number of 1

edit: C:\Users\Mahedi Munna\OneDrive\Desktop\Asseby\Shifting.asm

file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator

```
01 .model small
02 .stack 100h
03 .data
04 .code
05
06 main proc
07
08     mov cx,8
09     mov dh,'0'
10
11     mov ah, 1
12     int 21h
13     mov bl,al
14
15     level1:
16     shl bl,1
17     jnc zero
18
19     one:
20     mov dl, '1'
21     inc dh
22     jmp printDL:
23
24     zero:
25     mov dl, '0'
26     jmp printDL
27
28
29     printDL:
30     mov ah,2
31     int 21h
32     loop level1
33
34
35     mov dl,10
36     int 21h
37     mov dl,13
38     int 21h
39     mov dl,dh
40     int 21h
41
42     main endp
43 end main
```

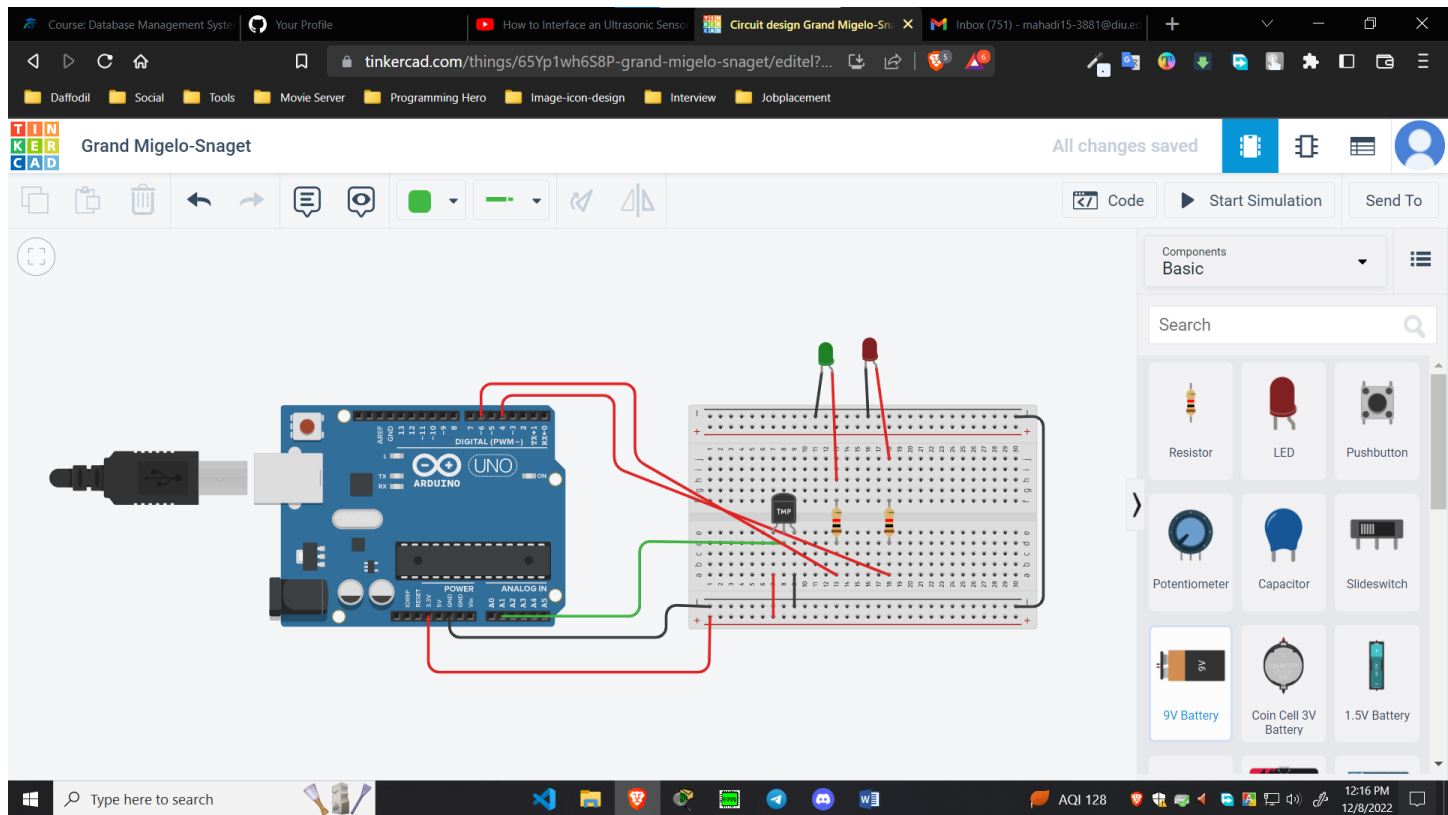
Output:

SCR emulator screen (80x25 chars)

300110011
4

Lab 7

1. Temperature Sensor:



Code:

```
void setup()
{
  pinMode(A1, INPUT);
  pinMode(6, OUTPUT);
  pinMode(4, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  float value = analogRead(A1);
  float voltage = value * (5/1024.0);
  float temp = (voltage - 0.5) * 100;
  Serial.print("Temp value: ");
  Serial.println(temp);
  if(temp>10) {
    if(temp<50) {
      digitalWrite(6, HIGH);
      digitalWrite(4, LOW);
    }
    else {
      digitalWrite(4,HIGH);
      digitalWrite(6,LOW);
    }
  }
}
```

```

else{
digitalWrite(4,HIGH);
digitalWrite(6,LOW);
}
delay(100);
}

```

