

emu8086 - assembler and microprocessor emulator 4.08

file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator convertor options

```

001
002 .model small
003 .stack 100h
004 .data
005
006 arr2 db 00110010b,01001001b,01001001b,01001001b,01001110b
007 arr0 db 00011100b,00100010b,01000001b,00100010b,00011100b
008 arr7 db 00000001b,01100001b,00010001b,00001101b,00000011b
009 arr1 db 01000010b,01000001b,01111111b,01000000b,01000000b
010
011
012 .code
013
014 main proc
015
016     mov ax,@data
017     mov ds,ax
018
019
020     mov si,0
021     mov ax,0
022     mov dx,2000h ; starting port address
023     mov cx,5
024
025     ;print 2
026 lp2:
027     mov al,arr2[si]
028     out dx,al
029     inc si
030     mov ax,0
031     add dx,1h
032
033     loop lp2
034
035
036     ;print 0
037     mov si,0
038     mov ax,0
039     mov cx,5
040 lp20:
041     mov al,arr0[si]
042     out dx,al
043     inc si
044     mov ax,0
045     add dx,1h
046
047     loop lp20
048
049

```

```

049
050     ;print 0
051
052     mov si,0
053     mov ax,0
054     mov cx,5
055     lp200:
056     mov al,arr0[si]
057     out dx,al
058     inc si
059     mov ax,0
060     add dx,1h
061
062     loop lp200
063
064
065     ;print 7
066     mov si,0
067     mov ax,0
068     mov cx,5
069     lp2007:
070     mov al,arr7[si]
071     out dx,al
072     inc si
073     mov ax,0
074     add dx,1h
075
076     loop lp2007
077
078
079     ;print 1
080     mov si,0
081     mov ax,0
082     mov cx,5
083     lp20071:
084     mov al,arr1[si]
085     out dx,al
086     inc si
087     mov ax,0
088     add dx,1h
089
090     loop lp20071
091
092
093     ;print 1
094     mov si,0
095     mov ax,0
096     mov cx,5
097     lp200711:
098     mov al,arr1[si]
099     out dx,al
100     inc si
101     mov ax,0
102     add dx,1h
103
104     loop lp200711
105

```

[illegible]

```

106
107
108    ;print 0
109    mov si,0
110    mov ax,0
111    mov cx,5
112    lp2007110:
113    mov al, arr0[si]
114    out dx, al
115    inc si
116    mov cx,0
117    add dx, 1h
118
119    loop lp2007110
120
121
122    exit:
123    mov ah, 4ch
124    int 21h

```

```
arr2 db 00110010b,01001001b,01001001b,01001001b,01001110b
arr0 db 00011100b,00100010b,01000001b,00100010b,00011100b
arr7 db 00000001b,01100001b,00010001b,00001101b,00000011b
arr1 db 01000010b,01000001b,01111111b,01000000b,01000000b
```

.code

main proc

mov ax,@data
mov ds,ax

mov si,0
mov ax,0
mov dx,2000h ; starting port address
mov cx,5

;print 2
lp2:
mov al,arr2[si]
out dx,al
inc si
mov ax,0
add dx,1h

loop lp2

;print 0
mov si,0
mov ax,0
mov cx,5
lp20:
mov al,arr0[si]
out dx,al
inc si
mov ax,0
add dx,1h

loop lp20

;print 0

mov si,0
mov ax,0
mov cx,5
lp200:
mov al,arr0[si]
out dx,al
inc si
mov ax,0
add dx,1h

loop lp200

;print 7
mov si,0
mov ax,0
mov cx,5
lp2007:

```
mov al,arr7[si]
out dx,al
inc si
mov ax,0
add dx,1h

loop lp2007
```

```
;print 1
mov si,0
mov ax,0
mov cx,5
lp20071:
mov al,arr1[si]
out dx,al
inc si
mov ax,0
add dx,1h

loop lp20071
```

```
;print 1
mov si,0
mov ax,0
mov cx,5
lp200711:
mov al,arr1[si]
out dx,al
inc si
mov ax,0
add dx,1h

loop lp200711
```

```
;print 0

mov si,0
mov ax,0
mov cx,5
lp2007110:
mov al,arr0[si]
out dx,al
inc si
mov ax,0
add dx,1h

loop lp2007110
```

```
exit:
mov ah,4ch
int 21h
```

```
main endp
end main
```