Microprocessor and Assembly Language

All Source Code - (Click Here)

```
.model small
.stack 100h
.data

msg1 db "1. Summation$"
msg2 db 010,013,"2. Subtract$"
msg3 db 010,013,"3. Coupling$"
msg4 db 010,013,"4. Hexadecimal to Decimal$"
msg5 db 010,013,"5. Decimal to Hexidecimal$"
msg6 db 010,013,"6. Even - Odd$"
msg7 db 010,013,"7. Largest Number$"
msg8 db 010,013,"8. Smallest Number$"
msg9 db 010,013,"9. Repeat$"
msgA db 010,013,"A. Reverse$"
msgB db 010,013,"B. AAA$"
msgC db 010,013,"C. Star Pattern$"
msgD db 010,013,"D. Array$"
ha db 010,013,"Enter your choice: $"
sms1 db 010,013,"Enter first number: $"
sms2 db 010,013,"Enter second number: $"
sms3 db 010,013,"Summation: $"
sms4 db 010,013,"Subtract: $"
sms5 db 010,013,"Yes.$"
sms6 db 010,013,"No.$"
sms7 db 010,013,"Enter a hexadecimal number: $"
sms8 db 010,013,"Decimal: $"
sms9 db 010,013,"Enter a decimal number: $"
sms10 db 010,013,"Hexadecimal: $"
sms11 db 010,013,"Enter a number: $"
sms12 db 010,013,"Even.$"
sms13 db 010,013,"Odd.$"
sms14 db 010,013,"Enter thired number: $"
sms15 db 010,013,"Largest number is: $"
sms16 db " -> $"
sms17 db 010,013,"Enter a string: $"
sms18 db 010,013,"Smallest number is: $"
sms19 db 010,013,"How many lines you print star: $"
a db ?
b db ?
```

```
c db ?
thank db 010,013,010,013,001,003," Thank you $"
star1 db 010,013,"1. Right Triangle Star Pattern$"
star2 db 010,013,"2. Mirrored Right Triangle Star Pattern$"
star3 db 010,013,"3. Inverted Right Triangle Star Pattern$"
star4 db 010,013,"4. Inverted Mirrored Right Triangle Star Pattern$"
star5 db 010,013,"5. Pyramid Star Pattern$"
star6 db 010,013,"6. Square Star Patterns$"
ar db 010,013,"Array: $"
array1 db 010,013,"1. Show array number$"
array2 db 010,013,"2. Summation two index$"
array3 db 010,013,"3. Input Numbers$"
array4 db 010,013,"4. Search Element$"
arr db 1,2,3,4,5
arrinput db 10 dub(?)
num db 010,013,"Enter numbers: $"
show db 010,013,"Show Numbers: $"
search db 010,013,"Enter search number: $"

.code
main proc
   mov ax,@data
   mov ds,ax

   lea dx,msg1
   mov ah,09h
   int 21h

   lea dx,msg2
   mov ah,09h
   int 21h

   lea dx,msg3
   mov ah,09h
   int 21h

   lea dx,msg4
   mov ah,09h
   int 21h

   lea dx,msg5
   mov ah,09h
   int 21h

   lea dx,msg6
   mov ah,09h
```

```
int 21h

lea dx,msg7
mov ah,09h
int 21h

lea dx,msg8
mov ah,09h
int 21h

lea dx,msg9
mov ah,09h
int 21h

lea dx,msgA
mov ah,09h
int 21h

lea dx,msgB
mov ah,09h
int 21h

lea dx,msgC
mov ah,09h
int 21h

lea dx,msgD
mov ah,09h
int 21h

lea dx,ha
mov ah,09h
int 21h

mov ah,1
int 21h
mov a,al

mov ah,2
mov dl,010
int 21h
mov dl,013
int 21h

cmp a,'1'
je an
```

```
cmp a,'2'
je bn
cmp a,'3'
je cn
cmp a,'4'
je dn
cmp a,'5'
je en
cmp a,'6'
je fn
cmp a,'7'
je gn
cmp a,'9'
je hn
cmp a,'A'
je inn
cmp a,'8'
je jnn
cmp a,'B'
je knn
cmp a,'C'
je lnn
cmp a,'D'
je mnn

mnn:
lea dx,array1
mov ah,09h
int 21h

lea dx,array2
mov ah,09h
int 21h

lea dx,array3
mov ah,09h
int 21h

lea dx,array4
mov ah,09h
int 21h

lea dx,ha
mov ah,09h
int 21h
```

```
mov ah,1
int 21h
mov bl,al

mov ah,2
mov dl,010
int 21h
mov dl,013
int 21h

cmp bl,'2'
je ar1
cmp bl,'1'
je ar2
cmp bl,'3'
je ar3
cmp bl,'4'
je ar4

ar4:
call arrayy4
jmp exit

ar3:
call arrayy3
jmp exit

ar2:
call arrayy2
jmp exit

ar1:
call arrayy1
jmp exit

lnn:
lea dx,star1
mov ah,09h
int 21h

lea dx,star2
mov ah,09h
int 21h

lea dx,star3
mov ah,09h
```

```
int 21h

lea dx,star4
mov ah,09h
int 21h

lea dx,star5
mov ah,09h
int 21h

lea dx,star6
mov ah,09h
int 21h

lea dx,ha
mov ah,09h
int 21h

mov ah,1
int 21h
mov bl,al

mov ah,2
mov dl,010
int 21h
mov dl,013
int 21h

cmp bl,'1'
je aaan
cmp bl,'2'
je bbbn
cmp bl,'6'
je cccn
cmp bl,'3'
je dddn
cmp bl,'4'
je eeen
cmp bl,'5'
je fffn

fffn:
call starr5
jmp exit

eeen:
```

```
call starr4
jmp exit

dddn:
call starr3
jmp exit

cccn:
call starr6
jmp exit

bbbn:
call starr2
jmp exit

aaan:
call starr1
jmp exit

knn:
call aaaa
jmp exit

jnn:
call smallest
jmp exit

inn:
call reverse
jmp exit

hn:
call repeat
jmp exit

gn:
call large_number
jmp exit

fn:
call even_odd
jmp exit

en:
call decimal
jmp exit
```

```
dn:
call hexa
jmp exit

cn:
call coupling
jmp exit

bn:
call subtract
jmp exit

an:
call summation
jmp exit

exit:
lea dx,thank
mov ah,09h
int 21h

mov ah,4ch
int 21h
main endp

summation proc
   lea dx,sms1
   mov ah,09h
   int 21h

   mov ah,1
   int 21h
   mov bl,al

   lea dx,sms2
   mov ah,09h
   int 21h

   mov ah,1
   int 21h
   mov bh,al

   add bl,bh
   sub bl,48
   lea dx,sms3
   mov ah,09h
   int 21h

   mov ah,2
   mov dl,bl
   int 21h
   ret
   summation endp

subtract proc
   lea dx,sms1
   mov ah,09h
   int 21h

   mov ah,1
   int 21h
   mov bl,al

   lea dx,sms2
   mov ah,09h
   int 21h

   mov ah,1
   int 21h
   mov bh,al

   sub bl,bh
   add bl,48

   lea dx,sms4
   mov ah,09h
   int 21h

   mov ah,2
   mov dl,bl
   int 21h
   ret
   subtract endp

coupling proc
   lea dx,sms1
   mov ah,09h
   int 21h

   mov ah,1
   int 21h
   mov bl,al

   lea dx,sms2
   mov ah,09h
   int 21h

   mov ah,1
   int 21h
   mov bh,al

   cmp bl,bh
   je k
   jne kk

   k:
   lea dx,sms5
   mov ah,09h
   int 21h
   ret

   kk:
   lea dx,sms6
   mov ah,09h
   int 21h
   ret
   coupling endp

hexa proc
   lea dx,sms7
   mov ah,09h
   int 21h

   mov ah,1
   int 21h
   mov bl,al

   lea dx,sms8
   mov ah,09h
   int 21h

   cmp bl,'9'
   jle k1
   jg k2

   k2:
```

```
   sub bl,17

   mov ah,2
   mov dl,'1'
   int 21h
   mov dl,bl
   int 21h
   ret

   k1:
   mov ah,2
   mov dl,bl
   int 21h
   ret

   hexa endp

decimal proc
   lea dx,sms9
   mov ah,09h
   int 21h

   mov ah,1
   int 21h
   mov bl,al
   int 21h
   mov bh,al

   lea dx,sms10
   mov ah,09h
   int 21h

   cmp bl,'0'
   je l1
   jne l2

   l2:
   add bh,17

   mov ah,2
   mov dl,bh
   int 21h
   ret

   l1:
   mov ah,2
```

```
    mov dl,bl
    int 21h
    mov dl,bh
    int 21h
    ret
    decimal endp

even_odd proc
    lea dx,sms11
    mov ah,09h
    int 21h

    mov ah,1
    int 21h
    mov dl,al

    mov bl,2
    div bl

    cmp ah,0
    je even
    jne odd
    jmp exit

    even:
    lea dx,sms12
    mov ah,09h
    int 21h
    ret

    odd:
    lea dx,sms13
    mov ah,09h
    int 21h
    ret
    even_odd endp

large_number proc
    lea dx,sms1
    mov ah,09h
    int 21h

    mov ah,1
    int 21h
    mov bl,al
```

```
lea dx,sms2
mov ah,09h
int 21h

mov ah,1
int 21h
mov bh,al

lea dx,sms14
mov ah,09h
int 21h

mov ah,1
int 21h
mov cl,al

lea dx,sms15
mov ah,09h
int 21h

cmp bl,bh
jg p1
jl p2

p2:
cmp bh,cl
jg bhh
jl cll

bhh:
mov ah,2
mov dl,bh
int 21h
ret

p1:
cmp bl,cl
jg bll
jl cll

bll:
mov ah,2
mov dl,bl
int 21h
ret
```

```
    cll:
    mov ah,2
    mov dl,cl
    int 21h
    ret
    large_number endp

repeat proc
    lea dx,sms11
    mov ah,09h
    int 21h

    mov ah,2
    mov dl,010
    int 21h
    mov dl,013
    int 21h

    j:
    mov ah,1
    int 21h
    mov bl,al

    cmp bl,013
    je re
    lea dx,sms16
    mov ah,09h
    int 21h

    mov ah,2
    mov dl,bl
    int 21h

    mov ah,2
    mov dl,010
    int 21h
    mov dl,013
    int 21h
    jmp j

    re:
    ret

    repeat endp

reverse proc
```

```
   lea dx,sms17
   mov ah,09h
   int 21h

   mov cx,0

   ei:
   mov ah,1
   int 21h
   mov bl,al

   cmp bl,013
   je jj
   push bx
   inc cx
   jmp ei

   jj:
   mov ah,2
   mov dl,010
   int 21h
   mov dl,013
   int 21h

   suno:
   pop dx
   int 21h
   loop suno

   ret
   reverse endp

smallest proc
   lea dx,sms1
   mov ah,09h
   int 21h

   mov ah,1
   int 21h
   mov bl,al

   lea dx,sms2
   mov ah,09h
   int 21h

   mov ah,1
```

```
int 21h
mov bh,al

lea dx,sms14
mov ah,09h
int 21h

mov ah,1
int 21h
mov cl,al

lea dx,sms18
mov ah,09h
int 21h

cmp bl,bh
jl pp1
jg pp2

pp2:
cmp bh,cl
jl bhhh
jg clll

bhhh:
mov ah,2
mov dl,bh
int 21h
ret

pp1:
cmp bl,cl
jl blll
jg clll

blll:
mov ah,2
mov dl,bl
int 21h
ret

clll:
mov ah,2
mov dl,cl
int 21h
ret
```

```
    smallest endp

aaaa proc
    lea dx,sms1
    mov ah,09h
    int 21h

    mov ah,1
    int 21h
    mov bl,al

    lea dx,sms2
    mov ah,09h
    int 21h

    mov ah,1
    int 21h

    mov ah,0

    add ax,bx
    aaa

    add ax,3030h

    mov bx,ax

    lea dx,sms3
    mov ah,09h
    int 21h

    mov ah,2
    mov dl,bh
    int 21h
    mov dl,bl
    int 21h

    ret
    aaaa endp

starr1 proc
    lea dx,sms19
    mov ah,09h
    int 21h

    mov ah,1
```

```
   int 21h
   mov bl,al
   sub bl,48

   mov ah,2
   mov dl,010
   int 21h
   mov dl,013
   int 21h

   mov bh,1

   kka:
   cmp bh,bl
   jg exitt

   mov cx,0
   mov cl,bh

   hha:
   mov ah,2
   mov dl,042
   int 21h
   loop hha

   mov ah,2
   mov dl,010
   int 21h
   mov dl,013
   int 21h

   inc bh
   jmp kka

   exitt:
   ret

   starr1 endp

starr2 proc
   lea dx,sms19
   mov ah,09h
   int 21h

   mov ah,1
   int 21h
```

```
mov a,al
sub a,48

mov ah,2
mov dl,010
int 21h
mov dl,013
int 21h

mov bl,a
mov bh,a
sub bl,1

mov b,1

ko:
cmp bl,0
je st
mov cx,0
mov cl,bl

space:
mov ah,2
mov dl,032
int 21h
loop space

dec bl

st:
cmp b,bh
jg exi

mov cx,0
mov cl,b

sta:
mov ah,2
mov dl,042
int 21h
loop sta

inc b

mov ah,2
mov dl,010
```

```
   int 21h
   mov dl,013
   int 21h
   jmp ko

   exi:
   ret
   starr2 endp

starr6 proc
   lea dx,sms19
   mov ah,09h
   int 21h

   mov ah,1
   int 21h
   mov a,al
   sub a,48

   mov ah,2
   mov dl,010
   int 21h
   mov dl,013
   int 21h

   mov bh,1

   koo:
   cmp bh,a
   jg iii

   mov cx,0
   mov cl,a

   sta1:
   mov ah,2
   mov dl,042
   int 21h
   loop sta1

   mov ah,2
   mov dl,010
   int 21h
   mov dl,013
   int 21h
   inc bh
   jmp koo


   iii:
   ret
   starr6 endp

starr3 proc
   lea dx,sms19
   mov ah,09h
   int 21h

   mov ah,1
   int 21h
   mov a,al
   sub a,48

   mov ah,2
   mov dl,010
   int 21h
   mov dl,013
   int 21h

   doggg:
   cmp a,0
   je dogg
   mov cx,0
   mov cl,a

   dog:
   mov ah,2
   mov dl,042
   int 21h
   loop dog

   mov ah,2
   mov dl,010
   int 21h
   mov dl,013
   int 21h

   dec a
   jmp doggg

   dogg:
   ret
   starr3 endp

starr4 proc
   lea dx,sms19
   mov ah,09h
   int 21h

   mov ah,1
   int 21h
   mov a,al
   sub a,48

   mov ah,2
   mov dl,010
   int 21h
   mov dl,013
   int 21h

   mov b,1

   kutt:
   cmp a,0
   je ber

   mov cx,0
   mov cl,b

   spa:
   mov ah,2
   mov dl,032
   int 21h
   loop spa

   mov cx,0
   mov cl,a

   sttt:
   mov ah,2
   mov dl,042
   int 21h
   loop sttt

   mov ah,2
   mov dl,010
   int 21h
```

```
   mov dl,013
   int 21h

   inc b
   dec a
   jmp kutt

   ber:
   ret
   starr4 endp

starr5 proc
   lea dx,sms19
   mov ah,09h
   int 21h

   mov ah,1
   int 21h
   mov a,al
   sub a,48

   mov ah,2
   mov dl,010
   int 21h
   mov dl,013
   int 21h


   mov bl,a
   mov c,bl
   mov bh,a
   sub bl,1

   mov b,1

   koo1:
   cmp c,0
   je exii
   cmp bl,0
   je stt
   mov cx,0
   mov cl,bl

   spacee:
   mov ah,2
   mov dl,032
   int 21h
   loop spacee

   dec bl

   stt:
   mov cx,0
   mov cl,b

   staa:
   mov ah,2
   mov dl,042
   int 21h
   loop staa

   inc b
   inc b
   dec c

   mov ah,2
   mov dl,010
   int 21h
   mov dl,013
   int 21h
   jmp koo1

   exii:
   ret
   starr5 endp

arrayy1 proc
   lea dx,ar
   mov ah,09h
   int 21h


   mov di,0

   ff:
   cmp arr[di],0
   je fff
   mov ah,2
   mov dl,arr[di]
   add dl,48
   int 21h
   inc di
```

```
    jmp ff

    fff:
    lea dx,sms3
    mov ah,09h
    int 21h

    mov cx,5

    mov di,0

    sum:
    cmp arr[di+1],0
    je exiii
    mov al,arr[di]
    add al,arr[di+1]
    add al,48

    mov ah,2
    mov dl,al
    int 21h

    inc di
    loop sum

    exiii:
    ret
    arrayy1 endp

arrayy2 proc
    lea dx,ar
    mov ah,09h
    int 21h

    mov di,0

    po:
    cmp arr[di],0
    je ro
    mov ah,2
    mov dl,arr[di]
    add dl,48
    int 21h
    inc di
    jmp po
```

```
   ro:
   ret
   arrayy2 endp

arrayy3 proc
   lea dx,num
   mov ah,09h
   int 21h

   mov di,0
   mov cx,0

   cici:
   mov ah,1
   int 21h
   cmp al,013
   je cucu
   mov arrinput[di],al
   inc di
   inc cx
   jmp cici

   cucu:
   lea dx,show
   mov ah,09h
   int 21h

   mov di,0

   ctct:
   mov ah,2
   mov dl,arrinput[di]
   int 21h
   inc di
   loop ctct

   doha:
   ret
   arrayy3 endp

arrayy4 proc
   lea dx,num
   mov ah,09h
   int 21h

   mov di,0
   mov cx,0

   huha:
   mov ah,1
   int 21h
   cmp al,013
   je cudu
   mov arrinput[di],al
   inc di
   inc cx
   jmp huha

   cudu:
   lea dx,search
   mov ah,09h
   int 21h

   mov ah,1
   int 21h
   mov bl,al

   mov di,0

   kida:
   cmp bl,arrinput[di]
   je peaeci
   inc di
   loop kida
   jmp nai

   nai:
   lea dx,sms6
   mov ah,09h
   int 21h
   ret

   peaeci:
   lea dx,sms5
   mov ah,09h
   int 21h
   ret
   arrayy4 endp
```