

1a. Search a key element.

```
.model small
.stack
.data
arr dw 0111h,0112h,0113h,0114h,0115h
len dw ($-arr)/2
key equ 0113h
msg1 db "key is found at"
res db "position",13,10,"$"
msg2 db "key not found!","$"
.code
mov ax,@data
mov ds,ax
mov bx,00
mov dx,len
mov cx,key
again:cmp bx,dx
ja fail
mov ax,bx
add ax,dx
shr ax,1
mov si,ax
add si,si
cmp cx,arr[si]
jae big
dec ax
mov dx,ax
jmp again
big:je success
inc ax
mov bx,ax
jmp again
success:add al,01
add al,'0'
mov res,al
lea dx,msg1
jmp disp
fail:lea dx,msg2
disp:mov ah,09h
int 21h
mov ah,4ch
int 21h
end
```

2) Reverse string and check palindrome or not

.model small

.stack

.data

str db 20 dup(?)

msg db 'enter the string:\$'

msg0 db 13,10,' :is the reverse string''\$'

msg1 db 13,10, "string is a palindrome!!!!!"

msg2 db 13,10, "string is not a palindrome!!!!!"

.code

mov ax,@data

mov ds,ax

mov si,00h

lea dx,msg

mov ah,09h

int 21h

gets:mov ah,01h

int 21h

mov str[si],al

inc si

mov ah,0

push ax

cmp al,0dh

jnz gets

lea dx,msg0

mov ah,09h

int 21h

mov cx,si

rev:pop ax

mov dl,al

mov ah,02h

int 21h

loop rev

dec si

dec si

mov di,si

mov si,00h

nc:mov al,str[si]

cmp al,str[di]

jnz fl_msg

inc si

dec di

jnz nc

mov ah,09h

mov dx,offset msg1

int 21h

jmp end_pr

fl_msg: mov ah,09h

mov dx,offset msg2

int 21h

end_pr:mov ah,4ch

int 21h

end

```
3)to find NCR
.model small
.stack
.data
n db 4
r db 2
ncr db 0
.code
mov ax,@data
mov ds,ax
mov al,n
mov bl,r
mov ncr,0
call encr
add ncr,30h
mov dl,ncr
mov ah,02h
int 21h
exit:mov ah,4ch
int 21h
encr proc
cmp bl,al
je gat1
cmp bl,1
je gat3
cmp bl,0
je gat1
dec al
cmp bl,al
je gat2
push ax
push bx
call encr
pop bx
pop ax
dec bx
push ax
push bx
call encr
pop bx
pop ax
ret
gat1:inc ncr
ret
gat2:inc ncr
gat3:add ncr,al
ret
encr endp
end
```

4) ARRANGE THE NUMBERS IN ASCENDING ORDER

```
.model small
.stack
.data
list db 07h,09h,08h,03h,06h
n dw $-list
msg db "the sorted array is::$"
.code
mov ax,@data
mov ds,ax
mov bx,n
dec bx
nextpass:mov cx,bx
mov si,00
nextcomp:mov al,list[si]
inc si
cmp al,list[si]
jb next
xchg al,list[si]
mov list[si-1],al
next:loop nextcomp
dec bx
jnz nextpass
lea dx,msg
mov ah,09h
int 21h
mov bx,n
mov si,00h
again:mov al,list[si]
add al,'0'
mov dl,al
mov ah,02h
int 21h
mov ah,02h
mov al,' '
inc si
dec bx
jnz again
mov ah,4ch
int 21h
end
```


5)display time and date

```
.model small
.stack
.data
msg db 'the current system time and date is:','$'
hrs db 00h
min db 00h
sec db 00h
.code
mov ax,@data
mov ds,ax
mov ah,2ch
int 21h
mov hrs,ch
mov min,cl
mov sec,dh
lea dx,msg
mov ah,09h
int 21h
mov al,hrs
call displ
call disp2
mov al,min
call displ
call disp2
mov al,sec
call displ
jmp day

displ proc
aam
add ax,3030h
mov bx,ax
mov dl,bh
mov ah,02h
int 21h
mov dl,bl
mov ah,02h
int 21h
ret
displ endp
disp2 proc
mov dl,':'
mov ah,02h
int 21h
ret
disp2 endp
day:mov ah,2ah
int 21h
mov al,dl
aam
mov bx,ax
call disp
mov dl, '/'
mov ah,02h
int 21h
year:mov ah,2ah
```

```
int 21h
add cx,0f830h
mov ax,cx
aam
mov bx,ax
call disp
mov ah,4ch
int 21h
disp proc
mov dl,bh
add dl,30h
mov ah,02h
int 21h
mov dl,bl
add dl,30h
mov ah,02h
int 21h
ret
disp endp
end
```

.MODEL SMALL

.STACK

.DATA

FNAME DB 'C:\NEWFILE.ASM',00h

SUCCESS DB 'FILE IS CREATED SUCCESSFULLY','\$'

FAILURE DB 'ERROR DURING FILE CREATION','\$'

.CODE

MOV AX,@DATA

MOV DS,AX

MOV CX,06

LEA DX,FNAME

MOV AH,3CH ;3CH IS DOS INTERRUPT TO CREATE FILE

INT 21H

JC FAIL

LEA DX,SUCCESS

JMP EXIT

FAIL:

LEA DX,FAILURE

EXIT:

MOV AH,09H

INT 21H

MOV AH,4CH

INT 21H

END

.MODEL SMALL

.STACK

.DATA

FILE DB 'C:\NEWFILE.ASM',00H

SUCCESS DB 'FILE DELETED
SUCCESSFULLY','\$'

FAILURE DB 'ERROR DURING FILE
DELETION','\$'

.CODE

MOV AX,@DATA

MOV DS,AX

LEA DX,FILE

MOV AH,41H

INT 21H

JC FAIL

LEA DX,SUCCESS

JMP EXIT

FAIL:

LEA DX,FAILURE

EXIT:

MOV AH,09H

INT 21H

MOV AH,4CH

INT 21H

END

Q. Write a C program in Keil

- click on Keil microvision file from desktop.
 - Go to project, select new microvision project, select desktop, and give the filename as appa and then save.
 - In search type microclip
 - Expand microclip and select AT99551 and click okay. And then select No.
- Target 1 will be created.

→ Go to file select new and type new program.

```
#include <AT9955X.h>
```

```
int main()
```

```
{
```

```
int a, b, c, d, e;
```

```
a = 5;
```

```
b = 2;
```

```
c = a + b;
```

```
d = a - b;
```

```
e = a * b;
```

```
}
```

→ Go to file select save as file name.c and save it.

→ Expand Target 1 right click on source group + then select add existing

files to group source.

- Now select filename "app.c" click on add and then click on close.
- Now go to project click on build target. To check the errors.
- Again go to project click on rebuild all target files.
- Now select debug, click on start stop debug selection, then say ok.
- To see the output press $fn + f11$

Output:

a :	5
b :	2
c :	7
d :	3
c :	10(A)

