

PGM 1A

```
.model small
printf macro msg
    mov ah,9
    lea dx,msg
    int 21h
endm
exit macro
    mov ah,4ch
    int 21h
endm

.data
    a dw 1111h,2222h,3333h,4444h,5555h
    n dw ($-a)/2
    key dw 3333h
    low_ dw ?
    mid_ dw ?
    high_ dw ?
    m1 db 'Successful search$'
    m2 db 'Unsuccessful search$'

.code
    mov ax,@data
    mov ds,ax
    mov low_,0
    mov ax,n
    dec ax
    mov high_,ax

check:  mov si,low_
        cmp si,high_
        jg fail
        add si,high_
        shr si,1
        mov mid_,si
        shl si,1
        mov ax,key
        cmp ax,a[si]
        jne less
        printf m1
        exit

less:   cmp ax,a[si]
        jg great
        mov ax,mid_
        dec ax
        mov high_,ax
        jmp check

great:  mov ax,mid_
        inc ax
        mov low_,ax
        jmp check

fail:   printf m2
        exit
end
```

PGM 2A

```
include c:\masm\rd.mac
include c:\masm\disp.mac
.model small
.data
    str db 10 dup(?)

.code

    mov ax,@data
    mov ds,ax

rd1:    mov si,0
        read
        cmp al,13
        je done
        mov str[si],al
        inc si
        jmp rd1

done:   mov cx,si
        mov si,0

        mov dl,10
        display

disp1:  mov dl,str[si]
        display
        inc si
        loop disp1

        mov ah,4ch
        int 21h
end
```

PGM 3A

```
.model small
.data
    a db 85h,95h,25h,45h,55h,15h,65h,45h
    n dw $-a

.code

    mov ax,@data
    mov ds,ax

    mov bx,n
    dec bx

np:    mov cx,bx
    mov si,0

ni:    mov al,a[si]
    inc si
    cmp al,a[si]
    jbe next
    xchg al,a[si]
    mov a[si-1],al

next:  loop ni
    dec bx
    jnz np

    mov ah,4ch
    int 21h
end
```

PGM 4A

```
.model small
printf macro msg
    lea dx,msg
    mov ah,9
    int 21h
endm
scanf macro str
    lea dx,str
    mov ah,10
    int 21h
endm

.data
    str1 db 100
           db 0
           db 100 dup(0)
    str2 db 100
           db 0
           db 100 dup(0)
    m1   db 10,13,"Length of str1: "
    l1   db ?,10,13,"$"
    m2   db 10,13,"Length of str2: "
    l2   db ?,10,13,"$"
    m3   db 10,13,"Equal$"
    m4   db 10,13,"Not equal$"
    m5   db "Enter str1: $"
    m6   db 10,13,"Enter str2: $"

.code

    mov ax,@data
    mov ds,ax
    mov es,ax

    printf m5
    scanf str1

    printf m6
    scanf str2

    mov cl,str1+1
    mov bl,str2+1

    add cl,30h
    add bl,30h

    mov l1,cl
    mov l2,bl

    sub cl,30h
    sub bl,30h

    printf m1
    printf m2

    cmp cl,bl
    je next

dm4:   printf m4
        jmp exit

next:  lea si,str1+2
        lea di,str2+2
        cld
        repe cmpsb
        jne dm4
        printf m3

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

PGM 5A

```
.model small
.data
    str db 'malayalam'
    n db $-str
    rstr db 10 dup(0)
    m1 db "Palindrome$"
    m2 db "Not a palindrome$"

.code

    mov ax,@data
    mov ds,ax
    mov es,ax

    mov cl,n
    dec cx
    mov di,cx
    inc cx

bak:
    mov ah,str[di]
    mov rstr[si],ah
    dec di
    inc si
    loop bak

    lea si,str
    lea di,rstr
    cld
    mov cl,n
    repe cmpsb
    je dm1

    lea dx,m2
    jmp disp

dm1:    lea dx,m1
disp:  mov ah,9
    int 21h

    mov ah,4ch
    int 21h
end
```

PGM 6A

```
.model small
.data
    msg db "What is your name?"
    nam db 50 dup(0)

.code

    mov ax,@data
    mov ds,ax

    mov si,0
    mov ah,1
bak:   int 21h
    mov nam[si],al
    inc si
    cmp al,13
    jnz bak
    mov byte ptr nam[si],'$'

    call clr
    call setc

    mov ah,9
    lea dx,msg
    int 21h

    mov ah,4ch
    int 21h

clr:   mov ax,0600h
    mov bh,7
    mov cx,0
    mov dx,2479h
    int 10h
    ret

setc:  mov ah,2
    mov bh,0
    mov dh,12
    mov dl,20
    int 10h
    ret
end
```

PGM 7A

```
.model small
.data
    n dw 5
    r dw 3
    ncr dw ?

.code

    mov ax,@data
    mov ds,ax

    mov ax,n
    mov bx,r

    call ncrpro

    mov ah,4ch
    int 21h
ncrpro:
    cmp bx,ax
    je res1

    cmp bx,0
    je res1

    cmp bx,1
    je resn

    dec ax
    cmp bx,ax
    je incr
    push ax
    push bx
    call ncrpro
    pop bx
    pop ax

    dec bx
    push ax
    push bx
    call ncrpro
    pop bx
    pop ax

    ret

res1:
    inc ncr
    ret

incr:
    inc ncr

resn:
    add ncr,ax
    ret
end
```

PGM 8A

```
.model small
.code

    mov ah,2ch
    int 21h

    mov al, ch
    call disp

    mov dl, ':'
    mov ah,2
    int 21h

    mov al,cl
    call disp

    mov dl, ':'
    mov ah,2
    int 21h

    mov al,dh
    call disp

    mov ah,4ch
    int 21h

disp    proc near
        aam
        add ax,3030h

        mov bx,ax
        mov ah,2

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

        ret
disp    endp
end
```


PGM 9A

```
.model small
.data
    m1 db 10,13,"Enter file name to create:$"
    m2 db 10,13,"Enter file name to delete:$"
    menu db "1.Create",10,13,"0.Delete"
           db 10,13,"Enter choice:$"
    err db 10,13,"Error$"
    f_name db 80
           db 0
           db 80 dup(0)

.code

    mov ax,@data
    mov ds,ax

    lea dx,menu
    mov ah,9
    int 21h

    mov ah,1
    int 21h

    cmp al,'1'
    je c_file
    cmp al,'0'
    je d_file

error:  lea dx,err
        mov ah,9
        int 21h
        jmp exit

c_file: lea dx,m1
        mov ah,9
        int 21h
        call read
        lea dx,f_name
        mov ah,3ch
        mov cx,0
        int 21h
        jc error
        jmp exit

d_file: lea dx,m2
        mov ah,9
        int 21h
        call read
        lea dx,f_name
        mov ah,41h
        int 21h
        jc error
        jmp exit

read:   mov ah,1
        lea si,f_name
bak:    int 21h
        cmp al,13
        jz done
        mov [si],al
        inc si
        jmp bak
done:   ret

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

PGM 10A

```
.model small
.data
    m1 db 10,13,"Enter row: $"
    m2 db 10,13,"Enter col: $"
    m3 db 10,13,"Press any key to stop$"
    row db ?
    col db ?

.code

    mov ax,@data
    mov ds,ax

    lea dx,m1
    mov ah,9
    int 21h

    call read
    mov row,al

    lea dx,m2
    mov ah,9
    int 21h

    call read
    mov col,al

    lea dx,m3
    mov ah,9
    int 21h

    mov ah,2
    mov dh,row
    mov dl,col
    int 10h

    mov ah,8
    int 21h

    mov ah,4ch
    int 21h
read:
    mov ah,1
    int 21h
    and al,0fh
    mov bl,al

    mov ah,1
    int 21h
    and al,0fh
    mov ah,bl

    aad

    ret
end
```