1.AVERAGE OF NUMBERS

```
AVERAGE OF NUMBERS
DATA SEGMENT
    ARRAY DB 1,4,2,3,8,6,7,5,9
    AVG DB ?
    MSG DB "AVERAGE = $"
ENDS

CODE SEGMENT
   ASSUME DS:DATA CS:CODE
START:
     MOV AX,DATA
     MOV DS,AX

     LEA SI,ARRAY
     LEA DX,MSG
     MOV AH,9
     INT 21H

     MOV AX,00
     MOV BL,9

     MOV CX,9
     LOOP1:
         ADD AL,ARRAY[SI]
         INC SI
     LOOP LOOP1

     DIV BL

     ADD AL,30H

     MOV DL,AL
     MOV AH,2
     INT 21H

     MOV AH,4CH
     INT 21H
ENDS
END START
```

2.COPY CHARACTERS

```
COPY CHARACTERS
DATA SEGMENT 100h
   msg DB "SRM AP"
   count EQU ($-msg)
ENDS
CODE SEGMENT
   ASSUME DS:DATA,CS:CODE,ES:DATA
START:
mov ax,DATA
mov ds,ax
mov es,ax
lea si,msg
mov di,0x0Ah
mov cx,count
cld
rep movsb
ENDS
END START
```

3.GCD

```
GCD
mov al,2
mov bl,5
mov ah,0
cmp al,bl
ja next
xchg al,bl
next:
mov bh,bl
div bl
cmp ah,0
je l1
mov al,bh
mov bl,ah
mov ah,0
jmp next
```

```
l1:
mov dl,bl
add dl,48
mov ah,2h
int 21h
```

4.LARGEST NUMBER

```
LARGEST NUMBER
DATA SEGMENT
   array db 0,1
ENDS

CODE SEGMENT
   ASSUME DS:DATA CS:CODE
START:
   MOV AX,DATA
   MOV DS,AX
   MOV AL,array[si]
   MOV SI,0
   MOV BL,1

   LOOP1:INC SI
        CMP AL,array[SI]
        JGE LOOP2
        MOV AL,array[SI]

   LOOP2:DEC BL
        JNZ LOOP1

   ADD AL,30H
   MOV DL,AL
   MOV AH,02h
   INT 21h
   hlt

ENDS
END START
```

5.LCM

```
LCM
a db 2
b db 3
mov al,a
mov bl,b
mov ah,0
mov bh,al
div bl
cmp ah,0
jz exit
loop loop1
loop1:
mov ah,0
mov al,bh
add al,a
mov bh,al
div bl
cmp ah,0
jz exit
loop loop1
exit:
mov dl,bh
add dl,48
mov ah,2
int 21h
```

6.MATCHING CHARACTERS

```
MATCHING CHARACTERS
DATA SEGMENT 100h
   msg db 'SRM AP'
   count equ ($ - msg)
   pass db ' '
```

```
DATA ENDS
CODE SEGMENT
ASSUME DS:DATA,CS:CODE
START:
   mov ax,data
   mov ds,ax
   lea si,msg
   lea di,pass
   mov cx,count
   loop1:
      mov  al,[si]
      cmp al,[di]
      je l2
      inc si
      loop loop1
   l1:
      mov dl,'N'
      mov ah,2h
      int 21h
      hlt
   l2:
      mov dl,'Y'
      mov ah,2h
      int 21h
      hlt
   CODE ENDS
END START
```

---

7. NO OF VOWELS

```
NO OF VOWELS
DATA SEGMENT 100h
   v db 'AEIOUaeiou'
   count equ ($ - v)
   msg db 'aeiou'
   len equ ($ - msg)
   print DB 'Number of Vowels : $'
   DATA ENDS

CODE SEGMENT
   ASSUME DS:DATA,CS:CODE
   START:
```

```
    MOV AX,DATA
    MOV DS,AX
    MOV BL,0
    MOV BH,0
    LEA DI,MSG
    LEA DX,PRINT
    MOV AH,9H
    INT 21H
    L1:
    LEA SI,V
    MOV CX,COUNT

    LOOP1:
    MOV AL,[SI]
    CMP AL,[DI]
    JE L2
    L:
    INC SI
    LOOP LOOP1
    INC BH
    CMP BH,LEN
    JE COMPLETED
    INC DI
    JMP L1

    L2:
    INC BL
    JMP L

   COMPLETED:
      MOV DL,BL
      add dl,48
      MOV AH,2H
      INT 21H

   CODE ENDS
END START
```

---

8. PRINTING 0 TO 9

PRINTING 0 TO 9

```
org 100h
.data
a db 0,1,2,3,4,5,6,7,8,9
.code
mov cx,10
mov si,0
loop1:
mov al,a[si]
mov dl,al
add dl,48
mov ah,2h
int 21h
INC si
loop loop1
Hlt
```

---

9. SORTING

```
DATA SEGMENT
   A DB 5,9,2,7,2,1
   COUNT EQU 6
DATA ENDS
CODE SEGMENT
   ASSUME CS:CODE, DS:DATA
START:      MOV AX,DATA
            MOV DS,AX
            MOV CX,COUNT
            MOV DX,CX
            DEC DX
            MOV SI,0
OUT_LOOP:   CMP DX,SI
            JZ NXT
            MOV AL,A[SI]
            MOV BL,A[SI+1]
            CMP AL,BL
            JA SWAP
            INC SI
            JMP OUT_LOOP
SWAP:       MOV A[SI],BL
            MOV A[SI+1],AL
            INC SI
            JMP OUT_LOOP
```

```
NXT:      MOV SI,0H
          SUB CX,1
          CMP CX,0
          JNZ OUT_LOOP
          MOV CX,COUNT
          MOV SI,0
PRI:      MOV DL,A[SI]
          ADD DL,48
          MOV AH,2H
          INT 21H
          INC SI
          LOOP PRI
CODE ENDS
END START
```