2. Factorial

.model small

data segment

n dw 3

data ends

code segment

assume cs:code,ds:data

start:mov ax,data

mov ds,ax

mov ax,0000h

mov ax,n

mov cx,n

dec cx

b:dec n

mul n

dec cx

jnz b

mov ah,4ch

int 21h

code ends

end start

3. 8 bit binary number

.model small

data segment

org 1000h

n1 db 19h

n2 db ?

data ends

code segment

assume cs:code,ds:data

start:mov ax,@data

mov ds,ax

mov al,n1

mov bl,al

shr al,01

xor al,bl

mov n2,al

mov ah,4ch

int 21h

code ends

end start

4. Fibonacci

.model small

data segment

org 4000h

num db 07

org 5000h

fibo db ?

data ends

code segment

assume cs:code,ds:data

main:mov ax,data

mov ds,ax

lea di,fibo

mov dl,num

mov al,00h

mov [di],al

add di,01h

daa

mov bl,01h

mov [di],bl

up:add al,bl

daa

add di,01h

mov [di],al

xchg al,bl

dec dl

jnz up

mov ah,4ch

int 21h

code ends

end main

5. 0’s and 1’s

.model small

data segment

n dw 1000h

zero dw 01 dup(?)

one dw 01 dup(?)

data ends

code segment

assume cs:code,ds:data

start:mov ax,data

mov ds,ax

mov ax,n

mov bx,zero

mov dx,one

mov cx,10h

l:

rol ax,1

jc o

inc bx

jmp nxt

o:

inc dx

nxt:dec cx

jnz l

mov zero,bx

mov one,dx

mov ah,4ch

int 21h

code ends

end start

7. alphabet

.model small

code segment

assume cs:code

start:mov cx,26

mov dx,65

b:

mov ah,2

int 21h

inc dx

loop b

mov ah,4ch

int 21h

code ends

end start

8. linear search

.model small

print macro msg

lea dx,msg

mov ah,09h

int 21h

endm

data segment

array dw 1111h,2222h,3333h,3344h,4455h

len equ($-array)

key dw 2221h

msg1 db "key found $"

msg2 db "key not found $"

data ends

code segment

assume cs:code,ds:data

start: mov ax,data

mov ds,ax

lea si,array

mov cx,len

mov bx,key

next:mov ax,[si]

cmp ax,bx

je found

inc si

dec cx

jnz next

jne nf

nf:print msg2

jmp exit

found:print msg1

exit:mov ah,4ch

int 21h

code ends

end start

9. bubble sort

.model small

data segment

org 1000h

list dw 30h,10h,20h

count equ 3

data ends

code segment

assume cs:code,ds:data

start:mov ax,data

mov ds,ax

mov dx,count-1

again0:mov cx,dx

mov si,offset list

again1:mov ax,[si]

cmp ax,[si+2]

jl pr1

xchg [si+2],ax

xchg [si],ax

pr1:add si,02

loop again1

dec dx

jnz again0

mov ah,4ch

int 21h

code ends

end start

10. palindrome

.model small

data segment

s db "apple"

l dw $-s

rs db 10 dup(?)

m1 db "palindrome $"

m2 db "not palindrome $"

data ends

code segment

assume cs:code,ds:data

start:mov ax,data

mov ds,ax

mov es,ax

mov cx,l

lea si,s

lea di,rs

add di,cx

dec di

b:mov al,[si]

mov [di],al

inc si

dec di

loop b

lea si,s

lea di,rs

mov cx,l

repe cmpsb

jne np

lea dx,m1

mov ah,09h

int 21h

jmp d

np:lea dx,m2

mov ah,09h

int 21h

d:mov ah,4ch

int 21h

code ends

end start

11. nCr

.model small

data segment

s db "apple"

l dw $-s

rs db 10 dup(?)

m1 db "palindrome $"

m2 db "not palindrome $"

data ends

code segment

assume cs:code,ds:data

start:mov ax,data

mov ds,ax

mov es,ax

mov cx,l

lea si,s

lea di,rs

add di,cx

dec di

b:mov al,[si]

mov [di],al

inc si

dec di

loop b

lea si,s

lea di,rs

mov cx,l

repe cmpsb

jne np

lea dx,m1

mov ah,09h

int 21h

jmp d

np:lea dx,m2

mov ah,09h

int 21h

d:mov ah,4ch

int 21h

code ends

end start

12. time and date

.model small

.stack

.data

m1 db "current time:$"

m2 db 10,13,"current date:$"

hr db ?

min db ?

s db ?

day db ?

month db ?

year dw ?

.code

mov ax,@data

mov ds,ax

mov ah,2ch

int 21h

mov hr,ch

mov min,cl

mov s,dh

mov ah,2ah

int 21h

mov day,dl

mov month,dh

mov year,cx

lea dx,m1

mov ah,09

int 21h

mov cl,hr

mov ch,0

call disp

mov dl,':'

mov ah,2

int 21h

mov cl,min

mov ch,0

call disp

mov dl,':'

mov ah,02

int 21h

mov cl,s

mov ch,0

call disp

lea dx,m2

mov ah,09

int 21h

mov cl,day

mov ch,0

call disp

mov dl,'/'

mov ah,02

int 21h

mov cl,month

mov ch,0

call disp

mov dl,'/'

int 21h

mov ah,02

int 21h

mov cx,year

call disp

mov ah,4ch

int 21h

disp proc

mov bx,0

n:mov al,bl

add al,1

daa

mov bl,al

jnc n1

add al,1

daa

mov bh,al

n1:loop n

mov dl,bl

and dl,0f0h

mov cl,4

shr dl,cl

add dl,30h

mov ah,02

int 21h

mov dl,bl

and dl,0fh

add dl,30h

int 21h

ret

disp endp

end