Q1:

```
[org 0x0100]
 jmp start
; subroutine to scroll up the screen
; take the number of lines to scroll as parameter
scrollup: push bp
 mov bp,sp
 push ax
 push cx
 push si
 push di
 push es
 push ds
 mov ax, 80 ; load chars per row in ax
 mul byte [bp+4] ; calculate source position
 mov si, ax ; load source position in si
 push si ; save position for later use
 shl si, 1 ; convert to byte offset
 mov cx, 2000 ; number of screen locations
 sub cx, ax ; count of words to move
 mov ax, 0xb800
 mov es, ax ; point es to video base
 mov ds, ax ; point ds to video base
 xor di, di ; point di to top left column
 cld ; set auto increment mode
 rep movsw ; scroll up
 mov ax, 0x0720 ; space in normal attribute
 pop cx ; count of positions to clear
 rep stosw ; clear the scrolled space
 pop ds
 pop es
 pop di
 pop si
 pop cx
 pop ax
 pop bp
 ret 2
start: mov ax,5
 push ax ; push number of lines to scroll
 call scrollup ; call the scroll up subroutine
 mov ax, 0x4c00 ; terminate program
 int 0x21
```

Q2:

```
[org 0x0100]

jmp start

msg: db ’I am From Pakistan’
length: dw 18

clrscr:
push es
push ax
push cx
push di

mov ax,0xb800
mov es,ax
xor di,di
mov ax,0x0720
mov cx,2000

cld
rep stosw

pop di
pop cx
pop ax
pop es
ret

printstr:
push bp
mov bp,sp
push es
push ax
push cx
push si
push di

mov ax,0xb800
mov es,ax
mov al,80
mul byte[bp+10]
add ax,[bp+12]
shl ax,1
mov di,ax
mov si,[bp+6]
mov cx,[bp+4]
mov ah,[bp+8]

cld
nextchar:
lodsb
cmp al,0x20
jz skip
stosw

skip:
loop nextchar

pop di
pop si
pop cx
pop ax
pop es
pop bp
```

```
ret 10

start:
call clrscr

mov ax,30
push ax
mov ax,20
push ax
mov ax,1
push ax
mov ax,msg
push ax
push word[length]
call printstr

mov ax,0x4c00
int 0x21
```