**ΑΣΚΗΣΗ 4**

Η εντολή CALL 3000H βρίσκεται στο μέσο της εκτέλεσής της όταν πραγματοποιείται η διακοπή, συνεπώς θα ολοκληρωθεί προτού κληθεί η ρουτίνα εξυπηρέτησης της διακοπής. Άρα ο (PC) θα λάβει την τιμή 3000H και η προηγούμενη τιμή του (PC) θα αποθηκευτεί στη στοίβα. Στην στοίβα θα εισαχθεί πρώτα το byte υψηλότερης τάξης, και εν συνεχεία το byte χαμηλότερης τάξης. Ο (SP) λαμβάνει την τιμή 3FFEH. Μόλις ολοκληρωθεί η εκτέλεση της εντολής αυτής, θα κληθεί να εξυπηρετηθεί η διακοπή που προέκυψε. Η ρουτίνα εξυπηρέτησης της διακοπής RST 5.5 βρίσκεται στη θέση 2CH συνεπώς ο (PC) θα λάβει την τιμή 002CH, ενώ η προηγούμενη τιμή του θα αποθηκευτεί ομοίως με πριν στην στοίβα. Ο (SP) τώρα θα έχει τιμή 3FFCH. Μόλις ολοκληρωθεί η ρουτίνα εξυπηρέτησης, ο (PC) παίρνει την τιμή των 2 τελευταίων καταχωρήσεων στην στοίβα, ώστε η εκτέλεση του προγράμματος να συνεχιστεί από εκεί που διακόπηκε. Άρα, ο (PC) θα έχει τιμή 3000H και ο (SP) θα έχει τιμή 3FFEH, όπως ακριβώς και πριν την κλήση της ρουτίνας εξυπηρέτησης.

**ΑΣΚΗΣΗ 3**

(i)

SWAP\_NIBLE MACRO Q

PUSH PSW

MOV A,Q

RLC

RLC

RLC

RLC

MOV Q,A

MOV A,H

MOV H,L

MOV L,A

POP PSW

ENDM

(ii)

FILL MACRO ADDR,L,K

PUSH PSW

PUSH B

PUSH H

LXI H,ADDR

MVI A,L

MVI B,K

MVI C,00H

CPI 01H

JC CASE\_ZERO

CASE\_ZERO: CMA ;0 becomes 255

INR C ;indicates that length is 255+1

FILL\_ADDR: MOV M,B ;store in memory

INX H ;next address

DCR A ;loops left

CPI 01H

JNC FILL\_ADDR ;if A isnt 0 continue looping

CMP C

MVI A,01H

JC FILL\_ADDR ;if A is 0 but we have an additional loop, loop once more

POP H

POP B

POP PSW

END M

(iii)

RHLL MACRO N

PUSH B

MOV C,A ;CY register must not be stored, so we dont push psw

MVI B,N

ROT\_EXT\_REG: MOV A,L

RAL

MOV L,A

MOV A,H

RAL

MOV H,A

DCR B

MOV A,B

CPI 00H

JNZ ROT\_EXT\_REG

MOV A,C

POP B

ENDM