2.4.2

MOV AL, N //muta valoarea lui N in registrul AL

LBL_WHILE: CMP AL, 0 // Instructiunea while urmata de intructiunea de comparare pentru a se vedea daca AL = 0 sau nu, in caz afirmatie se apeleaza JZ FINAL, care este conditia iesirii din bucla

JZ FINAL ;condiția de ieșire din buclă este N = 0 (AL = 0 în acest caz)

DEC N // decrementam pe N

MOV AL, N // mutam noua valoare a lui N in AL

JMP LBL_WHILE // Salt din nou la while pentru a repeta procedeul

FINAL: NOP ; no operation - instrucțiune dummy, fără effect

2.4.3

MOV CX, N // se muta in registrul contor CX valoarea N pentru a stii cat sa mearga loop-ul

LBL_FOR: MOV AL, X // se executa loop-ul apoi se muta in AL valoarea lui X

SHR AL, 1 ;împărțire la 2 realizată prin deplasare la dreapta cu o poziție

MOV X, AL //se muta in X rezultatul impartirii la 2 din AL

LOOP LBL_FOR // Se reia loopul de la inceput

2.5 Intrebari

1. Pentru a ridica la patrat numarul salvat in AL trebuie apelata instructiunea in forma aceasta:

MUL AL

2. MOV AL, X ; mutam in AL valoarea X

MUL AL ; facem patratul valorii respective

MUL X ; inmultim inca o data cu X valoarea din AL pentru a calcula cubul

3. SHL AL, 3 ; inmultire cu 8 realizati prin deplasare la stanga cu 3 pozitii

4.

Pentru rezultatul -5:

- ZF este 0 deoarece rezultatul nu e 0
- SF este 1 pentru ca bitul cel mai semnificativ e 1

Pentru rezultatul 0:

- ZF este 1 pentru ca rezultatul e 0
- SF este 0 pentru ca bitul cel mai semnificativ e 0

5. Este instructiunea de salt conditionat LOOP care decrementeaza automat pe CX daca si numai daca CX != 0

6. In urma executiei intructiunii daca cele doua valori din registre sunt egale atunci flag-ul ZF va fi pe 1 deoarece rezultatul ultimei intructiuni aritmetice executate de procesor e 0.

In cazul in care AL < BL rezultatul ultimei instructiuni aritmetici executate de procesor va fi mai mica ca < 0 astfel flag-ul SF va fi pe 1.

7. Se folosestea instructiunea de salt JZ.

Exemplu: JZ E0