

Name: Lekhan Kumawat

Roll No: 1906055

Branch: CSE-1

Course: CS5469 Microprocessors and Micro Controllers

Course Code: CS5469

Date: 14-11-21

— x — x —

1. Write a program to separate odd & even numbers.

→ MODEL SMALL.

DATA

ARRAY DB, 12H, 98H, 45H, 83H, 28H, 67H, 74H Initializing contents of array.

ARR-EVEN DB, 10 DUP(?) stores even numbers.

ARR-ODD DB 10 DUP(?) Array for odd numbers.

CODE

START:

MOV AX, @DATA Initialize Data Segment.

MOV DS, AX :

MOV CL, 0AH Initialize the Counter

XOR DI, DI CLEAR Contents of DI } Initialize

XOR SI, SI " " of SI }

LEA BP, ARRAY Load effective address of Array in base pointer

BACK: MOV AL, DS:[BP] Move value pointed by BP in AL.

AND AL, 01H Masking all bits except LSB.

JZ NEXT Jump to next if zero flag is 1.

Name: Lekhan Kumawat

Roll No: 1906055

Branch: CSE-I

Course: CS5469 Microprocessors and Micro Controllers

Course Code: CS5469

Date: 14-11-21

— x — x —

LEA BX, ARR-ODD Load effective address of ARR-ODD to BX.

MOV [BX + DI], AL Move contents of AL to memory.

INC DI Increment odd pointer.

JMP SKIP Unconditional jump to SKIP.

NEXT: LEA BX, ARR-EVN Load effective address of ARR-EVN

MOV [BX + SI], AL Mov contents of AL to ARR-EVN

INC SI Increment even pointer.

SKIP: INC BP Increment back pointer.

LOOP BACK Jump to BACK.

MOV AH, 4CH

INT 21H Interrupt

END START End of Program.

Name: Lakhun Kumawat

Roll No: 1906055

Branch: CSE-I

Course: CS5469 Microprocessors and Micro Controllers

Course Code: CS5469

Date: 14-11-21

— x — x —

2) Multiply two 16-bit numbers.

soln.
MODEL SMALL

STACK

DATA

MULTIPLICAND DW 00FFH : Initialize 1st Number (Multiplicand)

MULTIPLIER DW 00FFH : Initialize 2nd Number (multiplier)

PRODUCT DW 2DUP (0) : Will be storing result of multiplication.

CODE

START:

MOV AX, @DATA Initialise Data segments.

MOV DS, AX

MOV AX, MULTIPLICAND Move multiplicand to AX register

MOV MULTIPLIER Multiply MULTIPLIER with Contents of AX

MOV PRODUCT, AX Store the result (lower nibble) AX

MOV PRODUCT+2, DX Store the higher nibble from DX

MOV AH, 4CH

INT 21H Interrupt

END START End of program

Name: Lakhman Kumawat

Roll No.: 1906055

Branch: CSE-I

Course: CS5469 Microprocessors and Micro Controllers

Course Code: CS5469

Date: 14-11-21

— x — x —

3) Write a program to operate positive & Negative Numbers.

- MODEL SMALL

- DATA

ARRAY DB 12H, -98H, -45H, 83H, -34H, 75H, -47H. Initializing array

NEG DB 10 DUP(?) Array for storing +ve numbers.

POS DB 10 DUP(?) Array for storing +ve numbers

- CODE

START: MOV AX, @DATA

MOV DS, AX

MOV CL, DAH

XOR DI, DI

XOR SI, SI

LEA BP, ARRAY

BACK : MOV AL, DS:[BP] Get the Number

AND AL, 80H Mask all bits except MSB

JZ NEXT Jump to NEXT if zero flag = 1

LEA BX, NEG Load effective address of NEG to BX

MOV [BX+DI], AL

INC DI

JMP SKIP.

Name: Lakhman Kumarwarat

Roll No: 1906055

Branch: CSE-1

Course: CS5469 Microprocessors and Micro Controllers

Course Code: CS5469

Date: 14-11-21

— x — x —

NEXT: LEA BX, POS

MOV [BX+SI], AL

INC SI

SKIP INC BP

LOOP BACK

MOV AH, 4CH

INT 21H

END START

NOTE: Only those instructions are explained which are not discussed in earlier programs here.

4) Divide two 16-bit numbers

→ MODEL SMALL

→ DATA

W1 DW 9222H Initialize W1 by Dividend.

W2 DW 1111H Initialize W2 by Divisor

Q DW ? States the value of quotient.

R DW ? States the value of remainder.

→ CODE

START: MOV AX, @DATA Initialize Data Segment

Mov DS, AX

Name: Lekhan Kumawat

Roll No: 1906055

Branch: CSE-I

Course: CS5469 Microprocessors and Micro Controllers

Course Code: CS5469

Date: 14-11-21

— x — x —

MOV AX,W1 Get dividend into AX register.

MOV BX,W2 Get divisor into the BX register

DIV BX Divide the Contents of AX by BX

MOV Q,AX Stores quotient in Q.

MOV R,DX Stores Remainder in R.

MOV AH,4CH

INT 91H Interrupt

END START End of program.

5). Write a program to find logical ones & zeros in a given data.

→ MODEL SMALL.

DATA

NUM DB AH Initialize NUM by data.

ONES DB 0 To store ones

ZEROS DB 0 To store zeros.

CODE

START: MOV AX, @DATA Initialize Data Segments

MOV DS,AX

MOV AL, NUM Move Value of NUM to AL reg.

Name: Lakhman Kumarat

Roll No: 1906055

Branch: CSE-1

Course: CS5469 Microprocessors and Micro Controllers

Course Code: CS5469

Date: 14-11-21

— x — x —

MOV CX, 08H . Initialize the Counter

BACK: ROR AL, 1 Right rotate by 1 (Move MSB in carry flag)
 INC ZERINC Check for 0 or 1 in Carry flag.
 INC ONES If 1, Increment ones.
 JMP NEXT JNMP to NEXT

ZERINC: INC ZEROS If 0, Increment zeros.

NEXT: DEC CX Decrement the Counter.
 JNZ BACK Jump to back till Counter = 0.
 MOV AH, 4CH
 INT 21H Interrupt
 END START End of Program.

6) Write a program to find the largest No. from the given array.

→ MODEL SMALL.

→ STACK 100.

→ DATA.

NUM DB 12H, 27H, 01H, 36H, 76H Initialize the array
 LARGE DB(?) To store largest number.

Name: Lakhman Kumawat

Roll No.: 1906055

Branch: CSE-I

Course: CS5469 Microprocessors and Micro Controllers

Course Code: CS5469

Date: 14-11-21

—x—x—

CODE:

START: MOV AX, @DATA Initialize Data Segment.

MOV DS, AX

MOV CS, OS

MOV AL, 00H Initialize AL reg by 0.

LEA SI, NUM Load effective address of NUM.

Loop 1: CMP AL, [SI] Compare Contents of AL & NUM.

JNC LOOP2 If no carry jump to Loop2.

MOV AL, [SI] else move Contents of [SI] to AL.

Loop 2: INC, SI Increment pointer

DEC CL Decrement Counter

JNZ LOOP1 Jump to loop1 till Counter ≠ 0

MOV LARGE, AL Store the Contents of AL to LARGE

MOV AH, 4CH

INT 21H Interrupt

END START End of program

Name: Lakhman Kumawat

Roll No: 1906055

Branch: CSE-I

Course: CS5469 Microprocessors and Micro Controllers

Course Code: CS5469

Date: 14-11-21

— x — x —

7) Write a program to find the smallest number in the array

- SOLⁿ →
- MODEL SMALL
 - STACK 100
 - DATA

NUM DB 12H, 37H, 01H, 36H, 76H

SMALLEST DB (?)

• CODE

START: MOV AX, @DATA

MOV DS, AX

MOV CS, 03H

MOV AL, FFH Initialise AL by FFH

LEA SI, NUM

LOOP1: CMP AL,[SI] Compare Contents of AL & [SI]
 JC LOOP2 If Carry flag=1 jump to Loop2
 MOV AL,[SI] Else load [SI] to AL reg.

LOOP2: INC .SI Increment pointer

DEC CL

JNZ LOOP1

MOV SMALLEST ,AL

MOV AH, 4CH

INT 21H

Interrupt

END START

End of program

Name: Lekhan Kumawat

Roll No.: 1906055

Branch: CSE-I

Course: CS5469 Microprocessors and Micro Controllers

Course Code: CS5469

Date: 14-11-21

— x — x —

Q) Write a program to transfer of a String in forward direction.

Soln

- MODEL SMALL

- DATA

SRC DB "NIT PATNA" Source String

DST DB 15 DUP(?) Destination string

- CODE

START : MOV AX, @DATA Initialize Data Segment

MOV DS, AX

MOV ES, AX Initialize Extra Segment

LEA SI, SRC Pointer Initialization of SRC

LEA DI, DST Pointer Initialization of DST

MOV CX, 09H Set the Counter

CLD Clear the direction flag

REP MOVSB Transfers the String byte till CX=0

MOV AH, 4CH (Auto Increment SI, DI & Decrement CX)

INT 21H Interrupt

END START End of program.

Name: Lakhman Kumawat

Roll No.: 1906055

Branch: CSE-I

Course: CS5469 Microprocessors and Micro Controllers

Course Code: CS5469

Date: 14-11-21

— x — x —

Q) Write a program to search a character in string.

✓ MODEL SMALL

✓ STACK 100

• DATA

STRING DB "COLLEGE" ; String

CHAR DB 'E' ; Character to search

RESULT DB (?)

COUNT EQU 07H

Length of String

• CODE

START: MOV AX, @DATA

Initialize Data segment

MOV DS, AX

MOV CX, COUNT

Set the Counter

LEN SI, STRING

pointer to String

MOV AL, CHAR

Store character in AL reg.

BACK: CMP, [SI], AL

Compare [SI] & AL reg.

JE STROBE1

If equal : Jump to STROBE1

INC SI

Increment pointer SI

DEC CX

Decrement Counter CX

JNZ BACK

Jump to back till CX=0

JMP STROBE

Jump to STROBE

Name: Lakhman Kumarwarat

Roll No.: 1906055

Branch: CSE-I

Course: CS5469 Microprocessors and Micro Controllers

Course Code: CS5469

Date: 14-11-21

— x — x —

STROBE1: MOV AL,01H MOV 01 to AL seg.

MOV RESULT,AL MOV AL to RESULT.

JMP LAST Jump to LAST.

STROBE: MOV AL,00H

MOV RESULT,AL

MOV 001 to AL seg when character
not in String.

MOV AL to RESULT.

LAST: MOV AH,4CH

INT 21H

Interrupt

END START

End of program.

10.) Write a program to reverse string :-

- MODEL SMALL

- DATA

x DB "PATNA" Given String

z DW (?) String length (here 8)

y DB (?) DUP (' ') , 'B' Reversed String.

- CODE

- START: MOV AX, @DATA

Initialize Data Segment.

Name: Roll No.: Course:

Lakhan Kumarwut 1906055 C95469

Page No.	
Date	

MOV DS, AX

LEA SI, Z-1

Pointers to last character of string.

LEA DI, Y

Pointers to first char of Reversed

MOV CX, Z

Initialize Counter.

LI:

MOV AL, [SI]

Move [SI] to AL

Mov [DI], AL

Move AL to [DI]

DEC SI

Decrement SI pointer

INC DI

Increment DI pointer

DEC CX

Decrement Counter

JNZ LI

Repeat till CX ≠ 0

LEA DX, Y

Display the reversed String.

MOV AH, 4CH

INT 21H

Interrupt.

END START

End of program