# Name: Muhammad Maiz Nadeem

Reg. ID: SP21-BCS-052

# Question 1:

**For a number stored on Ax register , create a procedure that display it on screen in a number system whose base is 6 (For checking 27d in this number system will be displayed as 43).**

# Answer:

|  |  |
| --- | --- |
| 1 | DIS MACRO STR |
| 2 | MOV AH,09H |
| 3 | LEA DX,STR |
| 4 | INT 21H |
| 5 | ENDM |
| 6 |  |
| 7 | DATA SEGMENT |
| 8 | MSG2 *DB* "BASE 6 NUMBER IS : $" |
| 9 | STR1 *DB* 20 DUP('$') |
| 10 | STR2 *DB* 20 DUP('$') |
| 11 | NO *DW* 180 |
| 12 | LINE *DB* 10,13,'$' |
| 13 | DATA ENDS |
| 14 |  |
| 15 | CODE SEGMENT |
| 16 | ASSUME DS:DATA, CS:CODE |
| 17 | START: |
| 18 | MOV AX, DATA |
| 19 | MOV DS, AX |
| 20 | LEA SI, STR1 |
| 21 | MOV AX, NO |
| 22 | MOV BH, 00 |
| 23 | MOV BL, 6 |
| 24 | L1:   DIV BL |
| 25 | ADD AH, '0' |
| 26 | MOV *BYTE* PTR[SI], AH |
| 27 | MOV AH, 00 |
| 28 | INC SI |
| 29 | INC BH |
| 30 | CMP AL, 00 |
| 31 | JNE L1 |
| 32 |  |
| 33 | MOV CL, BH |
| 34 | LEA SI, STR1 |
| 35 | LEA DI, STR2 |
| 36 | MOV CH, 00 |
| 37 | ADD SI, CX |
| 38 | DEC SI |
| 39 |  |
| 40 | L2:   MOV AH, *BYTE* PTR[SI] |
| 41 | MOV *BYTE* PTR[DI], AH |
| 42 | DEC SI |
| 43 | INC DI |
| 44 | LOOP L2 |
| 45 |  |
| 46 | DIS LINE |
| 47 | DIS MSG2 |
| 48 | DIS STR2 |
| 49 | MOV AH, 4CH |
| 50 | INT 21H |
| 51 |  |
| 52 | CODE ENDS |
| 53 | END START |

Question 2:  
  
**Make a procedure that asks user to enter a number , in a number system of base 6, and store the value of corresponding number in register AX.**

**(Hint: Pseudo codes at the end of MUL and DIV slides can be used for both questions . Just need to replace 10 with appropriate number system base).**

# Answer:

|  |  |
| --- | --- |
| 1 | DIS MACRO STR |
| 2 | MOV AH,09H |
| 3 | LEA DX,STR |
| 4 | INT 21H |
| 5 | ENDM |
| 6 |  |
| 7 | ORG 100h |
| 8 |  |
| 9 | .DATA |
| 10 |  |
| 11 | MSG1   *DB* "ENTER A BASE 6 NUMBER: $" |
| 12 | MSG2   *DB* "BASE 10 NUMBER IS: $" |
| 13 | INPUT  *DB* 20 DUP('$') |
| 14 | ANS    *DB* 0 |
| 15 | SIZE   *DB* 0 |
| 16 | LEN    *DB* 0 |
| 17 | BASE   *DB* 6 |
| 18 | LINE   *DB* 10, 13, '$' |
| 19 |  |
| 20 | .CODE |
| 21 |  |
| 22 | MAIN PROC |
| 23 |  |
| 24 | DIS MSG1 |
| 25 |  |
| 26 | LEA SI, INPUT |
| 27 | MOV AH, 1 |
| 28 |  |
| 29 | L1:  INT 21H |
| 30 | SUB AL, 30H |
| 31 | MOV [SI], AL |
| 32 | INC SI |
| 33 | INC SIZE |
| 34 | CMP AL, 0DDH |
| 35 | JNE L1 |
| 36 |  |
| 37 | DEC SIZE |
| 38 | MOV AL, SIZE |
| 39 | MOV LEN, AL |
| 40 | XOR AX, AX |
| 41 | XOR DX, DX |
| 42 |  |
| 43 | LEA SI, INPUT |
| 44 |  |
| 45 |  |
| 46 |  |
| 47 | L2:  MOV AL, [SI] |
| 48 | INC SI |
| 49 | DEC SIZE |
| 50 | MOV DL, SIZE |
| 51 | CMP DL, 0 |
| 52 | JE L4 |
| 53 | L3:  MUL BASE |
| 54 | DEC DL |
| 55 | CMP DL, 0 |
| 56 | JNE L3 |
| 57 | L4:  ADD ANS, AL |
| 58 | CMP SIZE, 0 |
| 59 | JNE L2 |
| 60 |  |
| 61 |  |
| 62 | DIS LINE |
| 63 | DIS MSG2 |
| 64 |  |
| 65 | XOR AX, AX |
| 66 | XOR DX, DX |
| 67 | MOV AL, ANS |
| 68 |  |
| 69 |  |
| 70 | MOV CX, 0 |
| 71 | MOV DX, 0 |
| 72 | LABEL1: CMP AX, 0 |
| 73 | JE PRINT1 |
| 74 |  |
| 75 | MOV  BX, 10 |
| 76 | DIV  BX |
| 77 | PUSH DX |
| 78 | INC  CX |
| 79 | XOR  DX, DX |
| 80 | JMP LABEL1 |
| 81 |  |
| 82 | PRINT1: CMP CX, 0 |
| 83 | JE EXIT |
| 84 | POP DX |
| 85 | ADD DX, 48 |
| 86 | MOV AH, 2 |
| 87 | INT 21H |
| 88 | DEC CX |
| 89 | JMP PRINT1 |
| 90 | EXIT: |
| 91 |  |
| 92 | MAIN ENDP |
| 93 |  |
| 94 | RET |