MPL

EXP6. Write an assembly language program to convert Packed BCD to Unpacked BCD and also convert it into its ASCII value.

.MODEL SMALL

.STACK 100H

.DATA

N1 DB 25H

DI1 DB ?

DI2 DB ?

A1 DB ?

A2 DB ?

.CODE

START:

MOV AX,@DATA

MOV DS,AX

MOV AL,N1

AND AL,0FH

MOV DI1,AL

ADD AL,30H

MOV A1,AL

MOV BL,N1

AND BL,0F0H

MOV CL,04H

ROL BL,CL

MOV DI2,BL

ADD BL,30H

MOV A2,BL

EXIT:

MOV AH,4CH

INT 21H

END START

EXP5. Write an assembly language program for 16 bit addition

.MODEL SMALL

.STACK 100H

.DATA

N1 DW 6587H

N2 DW 1111H

SUM DW 2 DUP(0)

.CODE

START: MOV AX,@DATA

MOV DS,AX

MOV AX,N1

MOV BX,N2

MOV DX,0000H

ADD AX,BX

MOV SUM,AX

JNC EXIT

INC DX MOV SUM+1,DX

EXIT:MOV AH,4CH

INT 21H

END START

EXP7. Write an assembly language program for Program to move set of numbers from one memory block to another.

.MODEL SMALL

.STACK 100H

.DATA SOU DB 24H,12H,34H,81H,59H

DSE DB 5 DUP(0)

.CODE

START:

MOV AX,@DATA

MOV DS,AX

LEA SI,SOU

LEA DI,DSE

MOV CX,5

BACK:

MOV AL,[SI]

MOV [DI],AL

INC SI

INC DI

LOOP BACK

EXIT:

MOV AH,4CH

INT 21H

END START

EXP8.Write an assembly language program to count number of 1’sand 0’s in a given 8-bit number.

.MODEL SMALL

.STACK 100H

.DATA

NUMBER DW 5648H

ZERO DW 01 DUP(?)

ONES DW 01 DUP(?)

.CODE

START:

MOV AX,@DATA

MOV DS,AX

MOV AX,NUMBER

MOV BX,00H

MOV DX,00H

MOV CX,10H

LABEL1:

ROL AX,1

JC ONE

INC BX

JMP NEXT

ONE:

INC DX

NEXT:

DEC CX

JNZ LABEL1

MOV ZERO,BX

MOV ONES,DX

MOV AH,4CH

INT 21H

END START

EXP9. Write an assembly language program to find even and odd numbers from a given list.

DATA SEGMENT

X1 DB 10D, 'ENTER THE NUMBER : $'

X2 DB 10D, 'THE GIVEN NUMBER IS AN EVEN NUMBER$'

X3 DB 10D, 'THE GIVEN NUMBER IS AN ODD NUMBER$'

DATA ENDS

CODE SEGMENT

ASSUME CS: CODE, DS: DATA

START:

MOV AX, DATA

MAV DS, AX

LEA DX, X1

MOV AH, 09h

INT 21h

MOV AH, 01h

INT 21h

MOV BL,2

DIV BL

CMP AH,0 ;CHECK IF REMAINDER IS ZERO

JE EVEN\_NUM;If AH is zero,its an even number

JMP ODD\_NUM

EVEN\_NUM:

LEA DX,X2

MOV AH,09h

INT 21h

JMP LAST

ODD\_NUM:

LEA DX,X3

MOV AH,09h

INT 21h

LAST: MOV AH,4Ch

INT 21h

CODE ENDS

END START

EXP10A. Write an ALP to find whether a string is palindrome or not

DATA SEGMENT

MSG1 DB 10,13,'ENTER ANY STRING :- $'

MSG2 DB 10,13,'ENTERED STRING IS :- $'

MSG3 DB 10,13,'LENGTH OF STRING IS :- $'

MSG4 DB 10,13,'NO, GIVEN STRING IS NOT A PALINDROME $'

MSG5 DB 10,13,'THE GIVEN STRING IS A PALINDROME $'

MSG6 DB 10,13,'REVERSE OF ENTERED STRING IS :- $'

P1 LABEL BYTE

M1 DB 0FFH

L1 DB ?

P11 DB 0FFH DUP ('$')

P22 DB 0FFH DUP ('$')

DATA ENDS

DISPLAY MACRO MSG

MOV AH,9H

LEA DX,MSG

INT 21H

ENDM

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

START:

MOV AX,DATA

MOV DS,AX

DISPLAY MSG1

LEA DX,P1

MOV AH,0AH

INT 21H

DISPLAY MSG2

DISPLAY P11

DISPLAY MSG3

MOV DL,L1

ADD DL,30H

MOV AH,2

INT 21H

DISPLAY MSG6

LEA SI,P11

LEA DI,P22

MOV DL,L1

DEC DL

MOV DH,0

ADD SI,DX

MOV CL,L1

MOV CH,0

REVERSE:

MOV AL,[SI]

MOV [DI],AL

INC DI

DEC SI

LOOP REVERSE

DISPLAY P22

LEA SI,P11

LEA DI,P22

MOV CL,L1

MOV CH,0

CHECK: MOV AL,[SI]

CMP [DI],AL

JNE NOTPALIN

INC DI

INC SI

LOOP CHECK

DISPLAY MSG5

JMP EXIT

NOTPALIN: DISPLAY MSG4

EXIT:

MOV AH,4CH

INT 21H

CODE ENDS

END START

EXP10B. Write an ALP to find whether a string is palindrome or not : Reverse String

.MODEL SMALL

.STACK 100H

.data

m1 DB 10D,"Enter String : $"

m2 DB 10D,"Reverse String : $"

str1 DB 15 DUP('$')

str2 DB 15 DUP('$')

len DB 0000h

print macro xx

lea dx,xx

mov ah,09h

int 21h

endm

.code

start:

mov ax,@data

mov ds,ax

print m1

lea si,str1

loop1:

mov ah,01h ; take input(character) from system

int 21h

cmp al,0Dh ; value for enter

je skip

mov [si],al

inc si

inc len

jmp loop1

call reverse

skip:

mov al,24h ;ending of string($)

mov byte[si],al

reverse proc near ; proc-procedure

lea si,str1

lea di,str2 ;di -destn. index

mov cl,[len]

mov ch,00

add di,cx

dec di

loop3:

mov al,[si]

mov [di],al

inc si

dec di

loop loop3

print m2

print str2

ret

reverse endp

mov ah,4ch

int 21h

end start

EXP11. Write an assembly language program to find factorial of a numbers.

.MODEL SMALL

.STACK 100H

.DATA

N DB 4H

.CODE

START:

MOV AX,@DATA

MOV DS,AX

MOV CL,N

LOOP1:

MUL CL

DEC CL

JNZ LOOP1

MOV AH,4CH

INT 21H

END START