

ARQUITECTURA DE COMPUTADORAS

PRÁCTICO N° 4

Salto condicionales (ver más detalles en el set de instrucciones)

JZ ** Salta a la dirección ** si z=1 (si en resultado anterior dio 0)
JNZ ** Salta a la dirección ** si z=0 (si en resultado anterior es ≠ 0)
JS ** Salta a la dirección ** si s=1 (si en resultado anterior dio negativo)
JGE** Salta a la dirección ** si el primer operando es mayor o igual que el segundo (de la instrucción inmediata anterior.
JMP ** salta a la dirección ** sin ninguna condición (salto incondicional)

- 1- Realizar un programa que tome un dato de la posición de memoria 0200. Enmascare el medio byte superior, convierta a BCD y lo almacene en la posición 0210.

```
MOV AH, [0200]
AND AH, 0F
CMP AH, 09
JZ **
JS **
ADD AH, 06
** MOV [0210], AH
INT 20
```

Enmascarar significa hacer cero todos los bit a considerar. O sea que enmascarar el medio byte superior significa poner a cero los 4 bits más significativos de un dato. Esto lo logramos con la instrucción lógica AND el número en binario 00001111.

De esta forma en AH vamos a obtener el dato de la memoria 0200 con los cuatro primeros bits en cero y los otros cuatro el valor que tenían, no se modifican. Ahora a este número, en AH, hay que convertirlo a BCD.

Si en valor de AH es igual a 09 (el bit Z=1 o sea el resultado de la comparación es cero) por lo tanto "salta" a **. Si no es cero el otro salto condicional (JS) se fija si S=1 o sea si es el valor de AH es negativo, es decir si AH es menor que 09, si esto ocurre "salta" a **.

En caso de no ser cero y no ser menor que 9 le suma 6 a AH (convierte a BCD), guarda AH en la posición 0210 y termina.

- 2- Dado un banco 100 de datos ubicado a partir de la posición 0200 realizar un programa que recupere el mayor de ellos en la posición 0300.

```
MOV CX, 64
MOV BX, 0201
MOV AH, [0200]
*** MOV AL, [BX]
CMP AH, AL
JGE **
MOV AH, AL
** INC BX
LOOP ***
MOV [300], AH
INT 20
```

El programa va comparando un dato con siguiente, guarda el mayor (en AH) y lo compara con el que sigue. Cuando termina en AH obtiene el mayor y lo guarda en 0300.

- 3- Dados 100 datos ubicados a partir de la dirección 0500, realice un programa que guarde en la posición 0600 la cantidad de números pares y en 0601 la de números impares.
- 4- Dado un texto, de 500 caracteres, almacenado en código ASCII a partir de la posición 0200, realizar un programa que cuente la cantidad de palabras y guarde el resultado en la posición 0700.
- 5- Guardar en la posición de memoria 0200 el promedio de diez datos ubicados a partir de la posición 01F6.
- 6- Realice un programa que cuente los números positivos y negativos de un banco de 100 datos ubicados a partir de la posición 0300.

```
MOV CX, 64
MOV DX, 0000
MOV BX, 300
**** MOV AH, [BX]
AND AH, 80
JZ **
INC DL
JMP ***
** INC DH
*** INC BX
LOOP ****
INT 20
```

DH cantidad de números positivo- DL cantidad de números negativo
MOV DX, 0000 Se inicializa los contadores DH y DL a cero. La instrucción AND AH, 80 pone a cero todos los bits de AH excepto el más significativo que es el que nos dice el signo del número (1 negativo, 0 positivo) en caso de que sea uno incrementa el contador de negativos y si es 0 incrementa el de los positivos.

- 7- Ídem al punto anterior pero considerando que los datos son de 16 bits.
- 8- Dado un banco de 50 datos ubicados a partir de la posición 0300 pasarlos a todos a positivos.
- 9- Realice un programa que encuentre el mayor número de 20 datos ubicados a partir de la posición 0200, lo convierta a BCD y lo almacene en la posición 0300.
- 10- Dado un banco de 100 datos ubicados a partir de la posición 0500, almacenar en la posición 020A la cantidad de datos que posean su 5to bit en 1.
- 11- En el mismo banco de datos del problema anterior, almacenar a partir de la posición 020C los que son múltiplos de 5.