

1. Hacer una rutina que verifique si la cadena almacenada en PALABRA (DB de 10 bytes) constituye un Palíndromo. De ser así deberá retornar Z=1 de lo contrario Z=0. (Z = flag).

Palíndromo: Palabra o expresión que es igual si se lee de izquierda a derecha que de derecha a izquierda.

palindromo-proc

```
mov cx, 5
mov si, 1
mov di, 10
```

compara:

```
cmp palabra[si], palabra[di] ;palabra está definido en DATA como cadena de caracteres
jne distinto
inc si
dec di
loop compara
jmp fin
```

distinto:

```
xor cx,cx
```

fin:

```
ret
```

palindromo-endp

2. Realizar un programa en assembler que pida el ingreso de 20 caracteres por teclado. Los mismos deben almacenarse en un arreglo llamado cadena. Luego deberán mostrarse por pantalla los caracteres que sean mayores que el promedio de todos los ingresados y tengan el bit 0 y bit 1 distintos. Utilizar procedimientos.

Una idea: dividir la funcionalidad en el ingreso de los caracteres por teclado, el cálculo del promedio y el recorrido del arreglo para mostrar los caracteres adecuados.

.data

```
cadena db (20)
```

.code

```
call ingreso-proc
call mostrar-proc
mov ah,4ch
int 21h
```

ingreso-proc:

```
mov ah, 01h
mov cx, 20
mov di, 00
xor bx,bx
```

tecla:

```
int 21h
mov cadena[di], al
add bl, al
inc di
loop tecla
ret
```

ingreso-endp:

mostrar-proc:

```
mov al,bl ;lo paso porque toma a "al" como registro para dividir por defecto
mov cl,20 ;"cl" registro auxiliar para hacer la division
div cl ;el resultado queda en "al"
mov cx, 20
mov ah, 02h
xor di, di
```

compara:

```
mov dl, cadena[di]
and dl, 03h
cmp dl, 00h
```

```
jz seguir
cmp dl, 03h
jz seguir
cmp cadena[di], al
jng seguir
int 21h
seguir:
inc di
loop compara
ret
mostrar-endp:
end
```