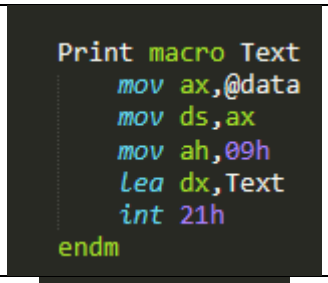
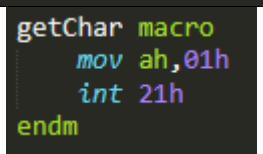
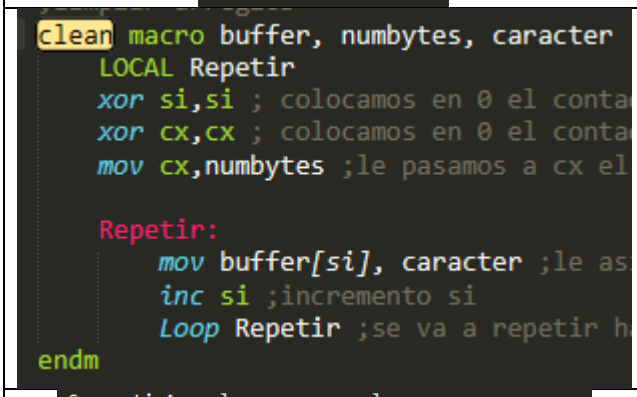
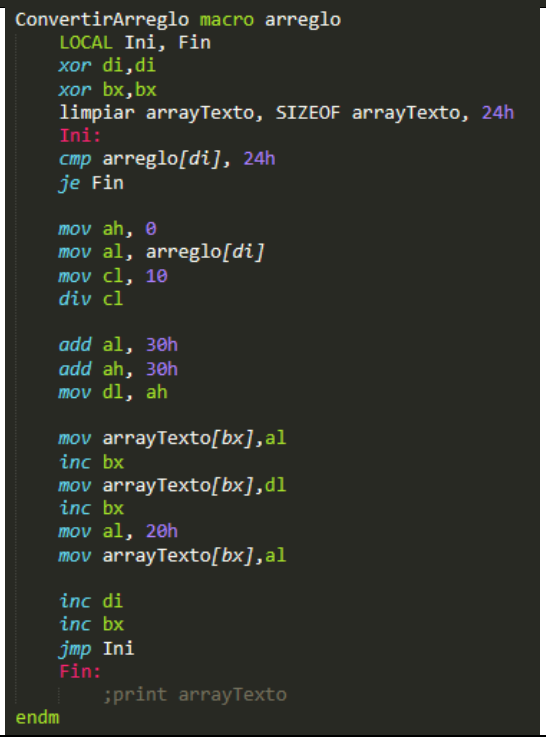
UNIVERSIDAD SAN CARLOS DE GUATEMALA
ARQUITECTURA DE COMPUTADORAS Y ENSAMBLADORES 1
SECCIÓN B
PRIMER SEMESTRE 2021

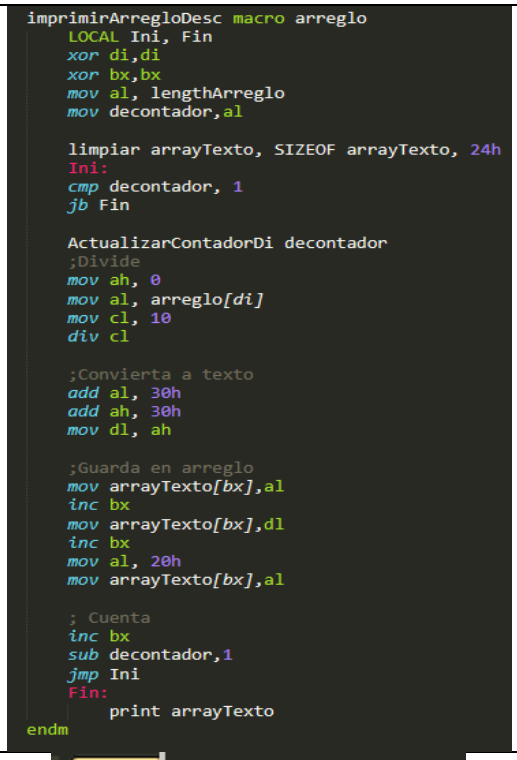
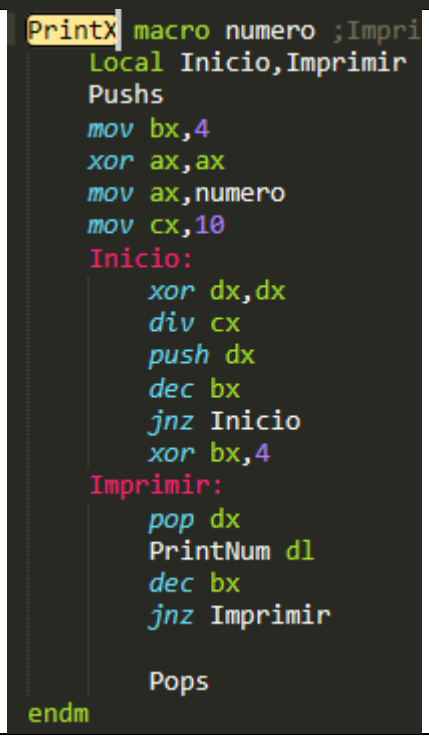
PROYECTO 4

201801263 Audrie Annelisse del Cid Ochoa

Manual Técnico

Macros utilizados constantemente a lo largo del proyecto:

Macro	Función
	Permite imprimir cadenas, utilizando la interrupción 21h
	Permite obtener un carácter. Se utilizó para obtener los datos al escoger una opción. Su valor queda guardado en al.
	clean, fue utilizado para limpiar arreglos. Permitía llenarlos con el carácter que se escogiera, en la mayoría de los casos fue el signo de dollar.
	Convertir Arreglo, fue utilizado para convertir un arreglo de enteros a uno de texto.

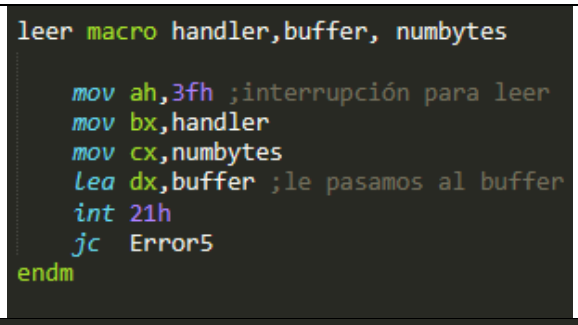
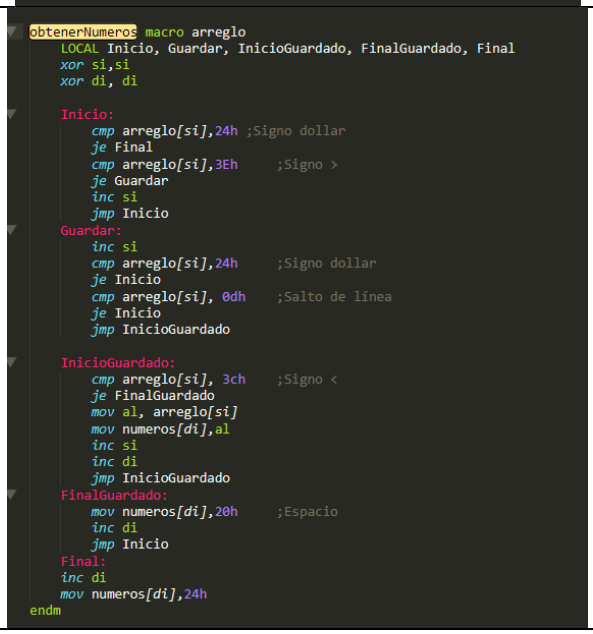
		Imprimir arreglo, fue utilizado para imprimir un arreglo de enteros.
		ImprimeDecimal, convierte los datos ascii en enteros y los imprime en pantalla.

1. Cargar Archivo

```
;je abrir
ComprobarArchivo bufferEntrada,NombreArchivo
je abrir
Comparar cmdmediana, bufferEntrada
```

Se utilizaron los siguientes Macros:

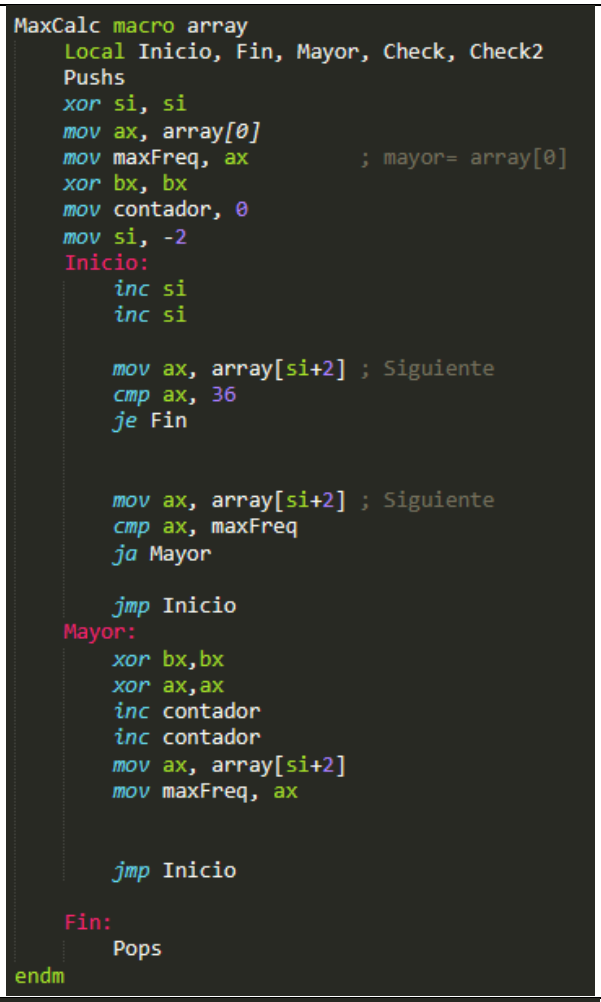
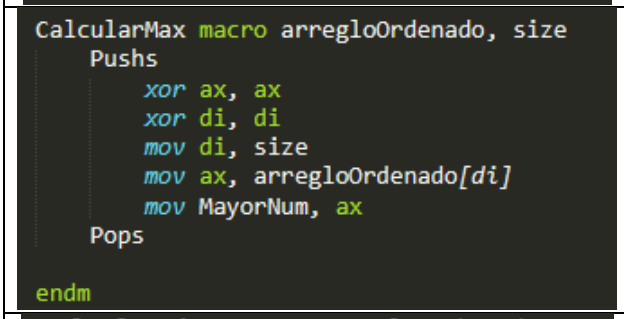
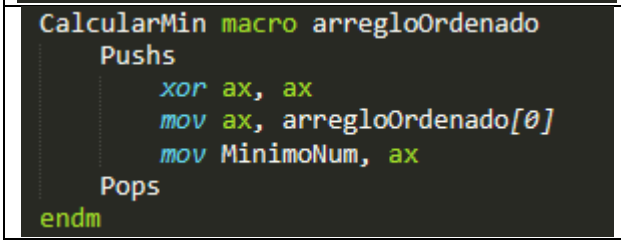
Macro	Función
`ComprobarArchivo macro entrada,nameArchivo` `local Inicio, Nombre,Fin,Salir,Comando` `Pushs` `mov nameArchivo,0` `xor di,di` `xor si,si` `Inicio:` `mov al,entrada[si]` `cmp al,"_"` `je Nombre` `cmp al,"$"` `je Fin` `inc si` `jmp Inicio` `Nombre:` `inc si` `mov al,entrada[si]` `cmp al,"$"` `je Comando` `mov nameArchivo[di],al` `inc di` `jmp Nombre` `Comando:` `mov nameArchivo[di+1],00h` `LeerArchivo nameArchivo` `Comparar cmdabrir,cmdabrir` `jmp Salir` `Fin:` `Comparar cmdsalir,cmdabrir` `jmp Salir` `Salir:` `Pops` `endm`	ComprobarArchivo, este macro analiza el texto ingresado y lo separa para comprobar que el comando es correcto, además de obtener el nombre del archivo para abrir.

		Leer, permite leer el arreglo de entrada los cuales contienen los datos que estaban dentro del documento que se ha abierto.
		Este macro fue utilizado para ir leyendo el arreglo de entrada que contiene los datos del documento y conforme va avanzando, los analiza y guarda en un arreglo de números ascii.

```
convertirNumero macro arreglo
    LOCAL Inicio, Guardar, Final
    mov contadorArreglo,0
    xor si,si
    xor di,di
    limpiar valorNumero, SIZEOF valorNumero, 24h

    Inicio:
        ;imprimir arreglo[si]
        cmp arreglo[si], 24h    ;Signo dollar
        je Final

        cmp arreglo[si],20h     ;Espacio
        je Guardar

        mov al, arreglo[si]
        mov valorNumero[di],al
        inc di
        inc si
        jmp Inicio
    Guardar:
        mov di, contadorArreglo
        mov auxsi, si
        mov auxdi, di
        obtenerSizeBuffer valorNumero
        filtarNumero valorNumero
        ;StringToInt valorNumero

        mov al, numero2
        ;imprimeDecimal numero2

        mov numeroReal[di],al
        limpiar valorNumero, SIZEOF valorNumero, 24h
        mov si, auxsi
        mov di, auxdi
        inc si
        xor di,di
        inc contadorArreglo
        jmp Inicio
    Final:
        ;imprimeDecimal numeroReal[0]
```

Este macro permite convertir los números del arreglo utilizado para guardar los números obtenidos del análisis del archivo cargado y los convierte en números decimales.

2. Funciones

Macro	Función
	MaxCalc, este macro permitía obtener el valor máximo de un arreglo. Se utilizó para obtener la moda, recorriendo el arreglo de frecuencias previamente obtenido
	CalcularMax, este macro permitía obtener el valor máximo de los datos guardados en un arreglo previamente ordenado.
	CalcularMin, , este macro permitía obtener el valor mínimo de los datos guardados en un arreglo previamente ordenado.

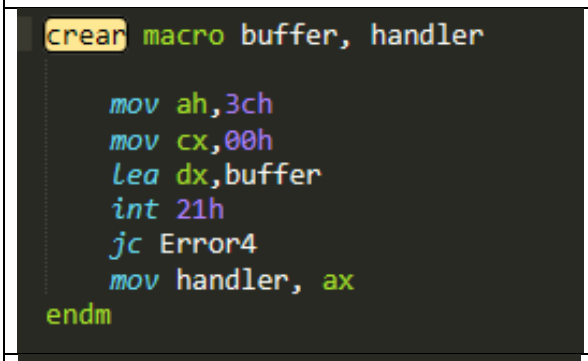
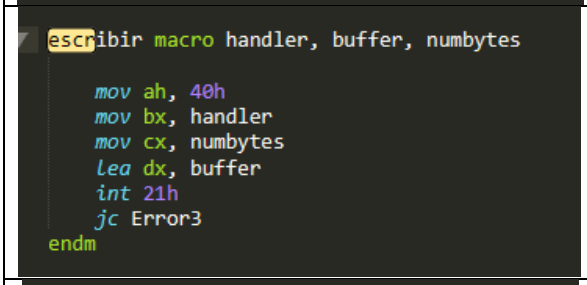
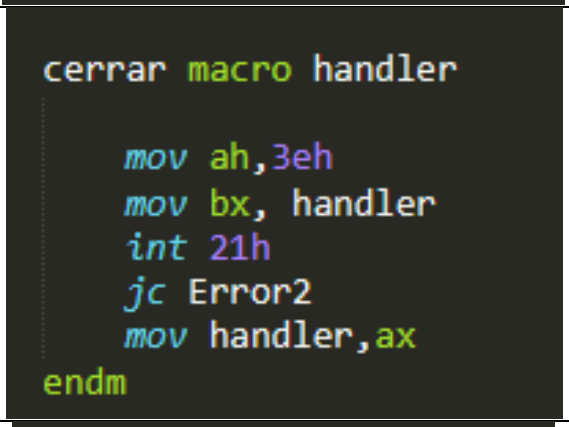
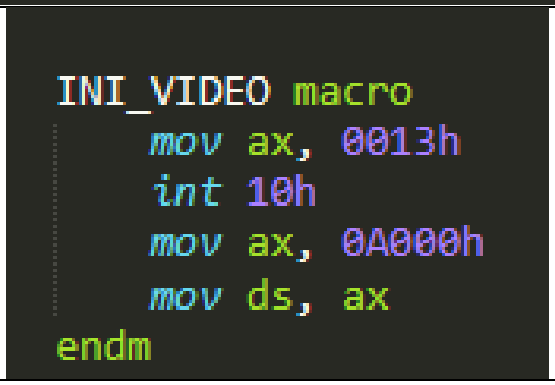
```
PromedioCalc macro arreglo
    Local Inicio, Dividir
    Pushs
    xor si, si
    Inicio:
        cmp si, lengthArray
        ja Dividir

        mov ax, arreglo[si]
        add sumaDatos, ax

        inc si
        inc si
        jmp Inicio
    Dividir:
        ;mov ax, sumaDatos
        ;cwd
        ;mov bx, TamaArreglo
        ;idiv bx
        ;mov promedio, ax
        ;mov residuo, dx
        ;jmp Fin
        ManejaDecimal sumaDatos, TamaArreglo, residuo, promedio
        ;Decimal_Text promedio, promedioTxt
        ;Decimal_Text residuo, residuoTxt
        Pops
endm
```

PromedioCalc, este macro permite calcular el promedio con los datos del archivo cargado al programa.

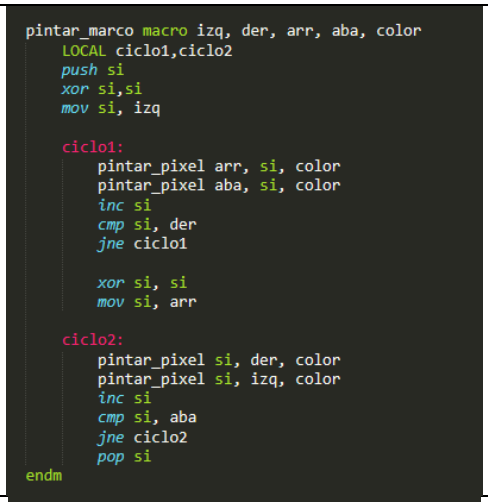
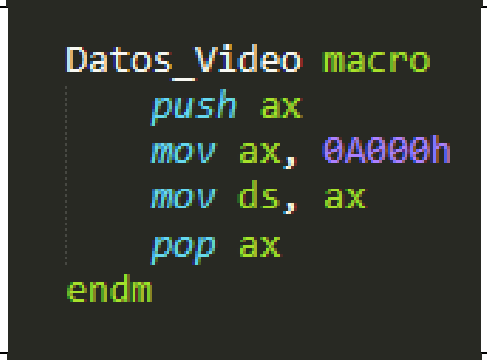
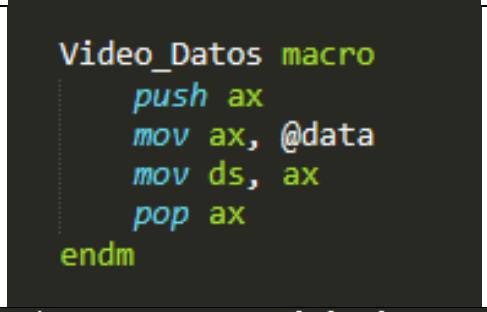
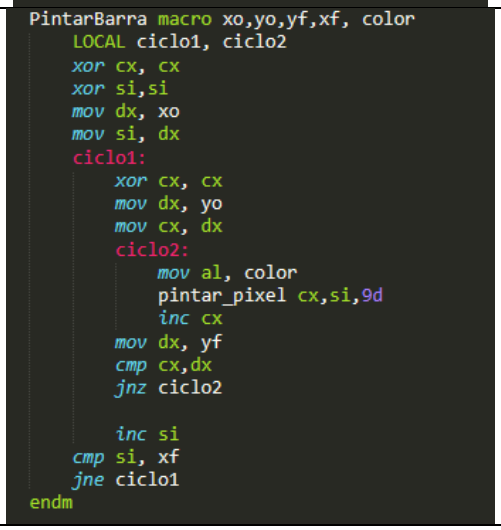
```
MedianaCalc macro arreglo, size
    Local Par, Impar, Fin
    mov aux2, 0
    mov aux, 0
    Pushs
    xor ax, ax
    xor bx,bx
    mov bx, 2
    mov ax, size
    add ax, 2
    cwd
    idiv bx
    mov Pivote, ax
    cwd
    idiv bx
    cmp dx, 0
    je Inicio
    jmp Impar
    Par:
        mov ax, size
        add ax, 2
        mov bx,2
        cwd
        idiv bx
        mov si,ax
        mov bx, arreglo[si]
        mov aux, bx
        dec si
        dec si
        mov bx, arreglo[si]
        mov aux2, bx
        mov ax, aux
        mov bx, aux2
        add ax, bx
        mov bx, 2
        mov aux, ax
        mov aux2, bx

        ManejaDecimal aux, aux2, MedDecimal, MedEntero
        jmp Fin
    Impar:
        mov ax, size
        mov bx, 2
        cwd
        idiv bx
        mov si, ax
        mov bx, arreglo[si]
        mov MedEntero, bx
        jmp Fin
    Fin:
        Pops
endm
```

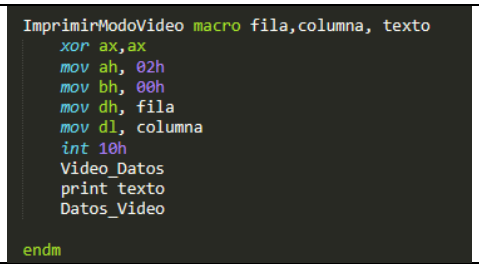
MedianaCalc, este macro permite calcular la mediana de los datos cargados al programa.

3. Generar Reporte

Macro	Función
	Permite crear un archivo.
	Permite escribir texto dentro del archivo creado y abierto.
	Permite cerrar el archivo abierto.
	Permite iniciar el modo video. Cambia de modo texto a modo video.

`` FIN_VIDEO macro mov ax, 0003h int 10h mov ax, @data mov ds, ax endm ```	Permite finalizar el mod video y cambia a modo texto
`` pintar_pixel macro a, b, color push ax push bx push di xor ax, ax xor bx, bx xor di, di mov ax, 320d mov bx, a mul bx add ax,b mov di, ax mov al, color mov [di],al pop di pop bx pop ax endm ```	Permite pintar un pixel en la pantalla, teniendo un lienzo de 320*200 pixeles. Se realiza mediante coordenadas y eligiendo un color.
`` delay macro param LOCAL ret2, ret1, finRet push ax push bx xor ax, ax xor bx, bx mov ax, param ret2: dec ax jz finRet mov bx, param ret1: dec bx jnz ret1 jmp ret2 finRet: pop bx pop ax endm ```	Delay, permite generar un retardo mediante la simulación de un bucle.

		Pintar_macro, permite pintar marcos mediante medidas de inicio en el eje x, fin en el eje x, inicio en el eje y , fin en el eje y. Así como el color.
		Datos_video, permite pasar de modo video a modo Texto, sin necesitad de sacarte de la aplicación.
		Video_Datos, permite pasar de modo video a modo texto sin sacarte de la aplicación.
		Pintar Barra, permite generar barras mediante coordenadas en diferentes posiciones del lienzo.

			Imprimir Modo Video, permite

4. Salir