

```
%macro print_int 1
    mov eax, 4
    mov ebx, 1
    mov ecx, %1
    mov edx, 4
    int 0x80
%endmacro
```

Se define un bloque de código que se llama print int1 que acepta un argumento, se preparan los registros para pedirle al OS que escriba algo en la pantalla.

```
section .data
    array dd 1, 2, 3, 4, 5

section .text
    global _start
```

Define las variables constantes se llama array, indica que los números ocupan 4 bytes con dd y contiene 1,2,3,4,5 en memoria consecutiva

```
_start:  
    mov ecx, 0  
    mov eax, 0  
  
bucle:  
    mov ebx, [array + ecx*4]  
    add eax, ebx  
  
    inc ecx  
    cmp ecx, 5  
    jl bucle  
  
    print_int eax  
  
    mov eax, 1  
    xor ebx, ebx  
    int 0x80
```

Inicia el programa y limpia los registros ecx es el índice para el arreglo y empieza en 0, mov eax es el acumulador donde se guardará la suma total y empieza en 0.
entra a la memoria y toma la dirección base del array, suma el índice ecx multiplicado por 4 y se carga el valor en ebx, luego se incrementa el índice en 1 y se compara el índice con 5 que es el tamaño del arreglo y salta si es menor, sí si es menor vuelve al bucle

En la salida se imprime eax
luego se termina el programa ejecutando la salida