

## Caso 1 TIC

Paula Carreño 202320149

Juan Andrés Moreno 202321829

### Análisis de pila:

**A partir del código C de la función `generate_tone` identifique las variables locales y parámetros.**

#### Parámetros:

**out:** short\*

**nSamples:** int

**freqHz:** unsigned int

**amp:** unsigned short

#### Variables Locales:

**phase:** unsigned int

**phaseStep:** unsigned int

**temp:** int

**sample:** short

**index:** unsigned char

**i=** int

**Pinte la pila justo antes de la ejecución de la primera instrucción de `generate_tone`, y justo después de la ejecución de la última instrucción (justo antes del retorno).**

**Justifique los desplazamientos asociados con variables locales y parámetros.**

Antes de ejecutar la primera instrucción

ESP + 16	amp
ESP + 12	freqHz
ESP + 8	nSamples
ESP + 4	out
ESP	return address

Pila justo antes del retorno

EBP + 8	out
EBP + 12	nSamples

EBP + 16	freqHz
EBP + 20	amp
EBP + 4	out
EBP	Viejo EBP

Completo:

EBP + 20	amp
EBP + 16	freqHz
EBP + 12	nSamples
EBP + 8	out
EBP + 4	dirección retorno
EBP	Viejo EBP
EBP - 4	phase
EBP - 8	phaseStep
EBP - 12	temp
EBP - 16	i
EBP - 20	reserva

Los desplazamientos se definen porque la función usa la pila para guardar parámetros y variables. Cuando la función empieza, EBP se usa como referencia fija. Los parámetros quedan en posiciones positivas desde EBP (EBP+8, EBP+12, etc.) porque antes del call se apilan y el call agrega la dirección de retorno. Cada parámetro ocupa 4 bytes en la pila, incluso si es un short, por cómo funciona la alineación en 32 bits. Las variables locales quedan en posiciones negativas (EBP-4, EBP-8, etc.) porque se reserva espacio restando a ESP con sub esp, Cada variable local tipo int ocupa 4 bytes, por eso se van acomodando de 4 en 4 hacia abajo en la pila.