```
extern printf
global main

section .data
format_specifier:
    db '%02x ', 0       ; Format specifier for hexadecimal with space
newline_format:
    db 10, 0          ; Newline format

section .text

main:
    push rbp
    mov rbp, rsp

    ; Print the first blank newline
    mov rdi, newline_format          ; Load newline-only format
    call printf                      ; Print an initial newline

    mov rbx, 1                       ; Set initial decrement step (1 for the first
line)

outer_loop:
    cmp rbx, 12                      ; Check if we’ve printed 13 lines
    jg end_program                   ; Exit if rbx > 12

    mov rcx, 0x18                    ; Initialize countdown value at 0x18 for each
line

inner_loop:
    push rbx                         ; Save rbx before calling printf
    push rcx                         ; Save rcx before calling printf

    mov rdi, format_specifier        ; Load format for hexadecimal output with
space
    mov rsi, rcx                     ; Move current countdown value to rsi for
printing
    call printf                      ; Print rcx in hexadecimal

    pop rcx                          ; Restore rcx after printf
    pop rbx                          ; Restore rbx after printf

    sub rcx, rbx                     ; Subtract the decrement step (rbx) from rcx
    cmp rcx, 0                       ; Check if rcx >= 0
    jge inner_loop                   ; Continue inner loop if rcx >= 0

    ; Print newline at the end of each line
    mov rdi, newline_format          ; Load newline format
    call printf                      ; Print newline

    inc rbx                          ; Increase the decrement step for the next
line
    jmp outer_loop                   ; Repeat outer loop for the next line

end_program:
    mov rax, 60                      ; syscall: exit
    xor rdi, rdi                     ; exit code 0
    syscall
```