```
;-----------------------------------------------------------------------------------
; Description: Fills an array of dword with N random ints in the range of [J, K]
; Author name: Koichi Nakata
; Author email: kanakta595@insite.4cd.edu
; Last modified date: March 28, 2024
; Creation date: March 28, 2024
;-----------------------------------------------------------------------------------

INCLUDE Irvine32.inc

.386
.model flat, stdcall
.stack 4096
ExitProcess PROTO, dwExitCode: dword

N = 100
J = 0
K = 49999999h

.data
array dword N dup(0)
msg1  byte  "First attempt: range of [0 to 49999999h]", 0
msg2  byte  "Second attempt: range of [49999999h to 99999999h]", 0

.code
main PROC
        mov  edx, offset msg1
        call WriteString

        mov  ecx, N
        mov  ebx, K - J + 1
        mov  edx, J
        mov  esi, offset array
        call fillArray
        call displayArray
```

```
        call Crlf
        mov  edx, offset msg2
        call WriteString

        J = 50000000h
        K = 99990000h
        mov  ebx, K - J + 1
        mov  edx, J
        call fillArray
        call displayArray

        INVOKE ExitProcess, 0
main ENDP

;-----------------------------------------------------------------------------
fillArray PROC
; Fills the array with N random ints [J, K]
; Receives: ECX: N, EBX: range(K-J+1), EDX: J, ESI: array pointer
; Returns: void
;-----------------------------------------------------------------------------
        pushad
        call Randomize                  ; Sets up a different seed
L1:
        mov  eax, ebx                   ; Sets up the range of random ints
        call RandomRange
        add  eax, edx                   ; Adds J
        mov  [esi], eax                 ; Stores the random int to the array
        add  esi, type array            ; Iterates to next element in the array
        loop L1

        popad
        ret
fillArray ENDP

;-----------------------------------------------------------------------------
displayArray PROC
```

```
; Displays the elements of the array
; Receives: void
; Returns: void
;-----------------------------------------------------------------------------
        pushad
        mov  esi, offset array
        mov  ecx, lengthof array
        mov  ebx, type array
        call DumpMem
        popad
        ret
displayArray ENDP

END main
```

```
;-----------------------------------------------------------------------------
; Description: Implements compound conditions OR, using conditional jmp
;               (ebx > ecx || ebx > val1) ? X = 1 : X = 2
; Author name: Koichi Nakata
; Author email: kanakta595@insite.4cd.edu
; Last modified date: March 28, 2024
; Creation date: March 28, 2024
;-----------------------------------------------------------------------------

INCLUDE Irvine32.inc

.386
.model flat, stdcall
.stack 4096
```

```
ExitProcess PROTO, dwExitCode: dword

.data
val1  dword 19191919
X     dword ?
msg1  byte  "X: ", 0
msg2  byte  "val1: ", 0
msg3  byte  "EBX: ", 0
msg4  byte  "ECX: ", 0

.code
main PROC
        mov  ebx, 19191920
        mov  ecx, 19191921

        cmp  ebx, ecx                   ; Evaluates ebx - ecx
        ja   L1                         ; Jump to L1 if the result is above 0
        cmp  ebx, val1                  ; Evaluates ebx - val1
        ja.  L1                         ; Jump to L1 if the result is above 0
        mov  X, 2                       ; Both conditions are false
        jmp  next                       ; Want to skip L1

L1:     mov  X, 1

next:
        call displayX
        INVOKE ExitProcess, 0
main ENDP

;-----------------------------------------------------------------------------
displayX PROC
; Displays the contents in X, val1, ebx and ecx
; Receives: void
; Returns: void
; Remarks: Save the value edx to the stack
;-----------------------------------------------------------------------------
```

```
        mov  edx, offset msg1
        call WriteString
        mov  eax, X
        call WriteDec
        call Crlf

        mov  edx, offset msg2
        call WriteString
        mov  eax, val1
        call WriteDec
        call Crlf

        mov  edx, offset msg3
        call WriteString
        mov  eax, ebx
        call WriteDec
        call Crlf

        mov  edx, offset msg4
        call WriteString
        mov  eax, ecx
        call WriteDec
        ret
displayX ENDP

END main
```

```
;-----------------------------------------------------------------------------
; Description: Implements compound conditions And and OR, using conditional jmp
;              ((ebx > ecx && ebx > edx) || edx > eax) ? X = 1 : X = 2
; Author name: Koichi Nakata
; Author email: kanakta595@insite.4cd.edu
; Last modified date: March 28, 2024
; Creation date: March 28, 2024
;-----------------------------------------------------------------------------

INCLUDE Irvine32.inc

.386
.model flat, stdcall
.stack 4096
ExitProcess PROTO, dwExitCode: dword

.data
X dword ?
msg byte "X: ", 0


.code
main PROC
        mov  eax, 11111111h
        mov  ebx, 22222222h
        mov  ecx, 10000000h
        mov  edx, 10000000h

        cmp  ebx, ecx                   ; Evaluates 1st condition
        jbe  L1                         ; Jump to L1 if False
        cmp  ebx, edx                   ; Evaluates 2nd condition
        jbe  L1                         ; Jump to L1 if False
        jmp  L2                         ; No need to evaluate 3rd condition if 1st and 2nd True
        jmp  next
```

```
L1:                                     ; Still have a chance to be True
        cmp  edx, eax                   ; Evaluates 3rd condition
        ja   L2                         ; Jump to L2 if True
        mov  X, 2                       ; Comes to this line if False
        jmp  next

L2:     mov  X, 1                       ; Comes to this block if True
        jmp  next

next:
        call DumpRegs
        mov  edx, offset msg
        call WriteString
        mov  eax, X
        call WriteDec
        INVOKE ExitProcess, 0
main ENDP
END main
```