.globl conv\_arr

conv\_arr:

pushq %rdi # \*x

pushq %rsi # n

pushq %rdx # \*h

pushq %rcx # m

movq $0, %r10 # i = 0

movq 16(%rsp), %r11 #r11 = n

addq (%rsp), %r11 #r11 = n + m

subq $2, %r11 #r11 = n + m - 2

loop:

cmp %r10, %r11 # size - i ? 0

jl endl # jump to endl if size-i < 0

test1:

movq %r10, %rdi #rdi = i

incq %rdi #rdi = i + 1

movq (%rsp), %rsi #rsi = m

call min

movq %rax, %r12 #r12 = ladj

test2:

movq (%rsp), %rdi #rdi = m

addq 16(%rsp), %rdi #rdi = m + n

subq %r10, %rdi #rdi = m + n - i

subq $1, %rdi #rdi = m + n - i - 1

movq (%rsp), %rsi #rsi = m

call min

movq %rsi, %r13 #r13 = m

subq %rax, %r13 #r13 = radj

test3:

movq 24(%rsp), %rdi #rdi = x

incq %rdi #rdi = x + 1

addq %r10, %rdi #rdi = x + 1 + i

subq %r12, %rdi #rdi = x + 1 + i -ladj

movq 8(%rsp), %rsi #rsi = h

addq %r13, %rsi #rsi = h + radj

movq %r12, %rdx #rdx = ladj

subq %r13, %rdx #rdx = ladj - radj

call conv

movb %al, (%r8, %r10) #result[i] = rax

incq %r10 #i = i + 1

jmp loop #jump to loop

endl:

popq %rcx

popq %rdx

popq %rsi

popq %rdi

ret