.globl conv\_arr

conv\_arr:

pushq %rdi #x = 24(%rsp)

pushq %rsi #n = 16(%rsp)

pushq %rdx #h = 8(%rsp)

pushq %rcx #m = (%rsp)

movq $0, %r9 #i(r9) = 0

movq 16(%rsp), %r10 #r10 = n

addq (%rsp), %r10 #r10 = n+m

subq $2, %r10 #r10 = n+m-2

loop:

cmpq %r10, %r9 #i - (n+m-2) ? 0

jg endl #end if i > (n+m-2)

movq %r9, %rdi #rdi = i

addq $1, %rdi #rdi = i+1

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

call min

movq %rax, %r11 #ladj(r11) = min(i+1, m)

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

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

subq %r9, %rdi #rdi = m + n - i

decq %rdi #rdi = m + n - i - 1

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

call min

movq (%rsp), %r12 #r12 = m

subq %rax, %r12 #radj(r12) = m - min((m+n-i-1),m)

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

addq %r9, %rdi #rdi = x + i

incq %rdi #rdi = x + i + 1

subq %r11, %rdi #rdi = x + i + 1 - ladj

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

addq %r12, %rsi #rsi = h + radj

movq %r11, %rdx #rdx = ladj

subq %r12, %rdx #rdx = ladj - radj

call conv

movb %al, (%r8, %r9) #result[i] = conv(x + i + 1 - ladj, h + radj,

ladj - radj)

incq %r9 #i++

jmp loop #back to loop

endl:

popq %rcx

popq %rdx

popq %rsi

popq %rdi

ret