## Source code

```
#let add(/*int*/x, /*int*/y) = x + y

2 + 2 = #add(2, 2)
```

## Compiled

```
segment .data
_P__P_const_0_entry: db ``, 0
_P__P_const_1_entry: db `\n\n2 + 2 = `, 0
_P__P_const_2_entry: db `\n`, 0

segment .text
    global entry

    extern cast_int_to_content
    extern content_join
    extern mk_function
_A_add_2:
    push rbp
    mov rbp, rsp
    sub rsp, 16
    mov [rbp - 8], rdi
    mov [rbp - 16], rsi
    mov rdx, [rbp - 8]
    mov rcx, [rbp - 16]
    add rdx, rcx
    mov rax, rdx
    mov rsp, rbp
    pop rbp
    ret
entry:
    push rbp
    mov rbp, rsp
    sub rsp, 96
    mov rdi, _P__P_const_0_entry
    mov [rbp - 16], rdi
    xor rax, rax
    mov rdi, _A_add_2
    mov rsi, 0
    call mk_function
    mov [rbp - 8], rax
    mov rsi, _P__P_const_1_entry
    mov [rbp - 24], rsi
    xor rax, rax
    mov rdi, [rbp - 16]
    mov rsi, [rbp - 24]
    call content_join
    mov rdx, [rbp - 8]
    mov rcx, 2
    mov r8, 2
    mov [rbp - 32], rbx
    mov [rbp - 40], rax
    mov [rbp - 48], rdx
    mov [rbp - 56], rcx
    mov [rbp - 64], r8
    xor rax, rax
    mov rdi, [rbp - 56]
    mov rsi, [rbp - 64]
    mov rbx, [rbp - 48]
    mov rbx, [rbx]
    call rbx
    mov [rbp - 72], rax
    xor rax, rax
    mov rdi, [rbp - 72]
    call cast_int_to_content
    mov [rbp - 80], rax
    xor rax, rax
    mov rdi, [rbp - 40]
    mov rsi, [rbp - 80]
    call content_join
    mov r9, _P__P_const_2_entry
    mov [rbp - 88], rax
    mov [rbp - 96], r9
    xor rax, rax
    mov rdi, [rbp - 88]
    mov rsi, [rbp - 96]
    call content_join
    mov rax, rax
    mov rsp, rbp
    pop rbp
    ret
```