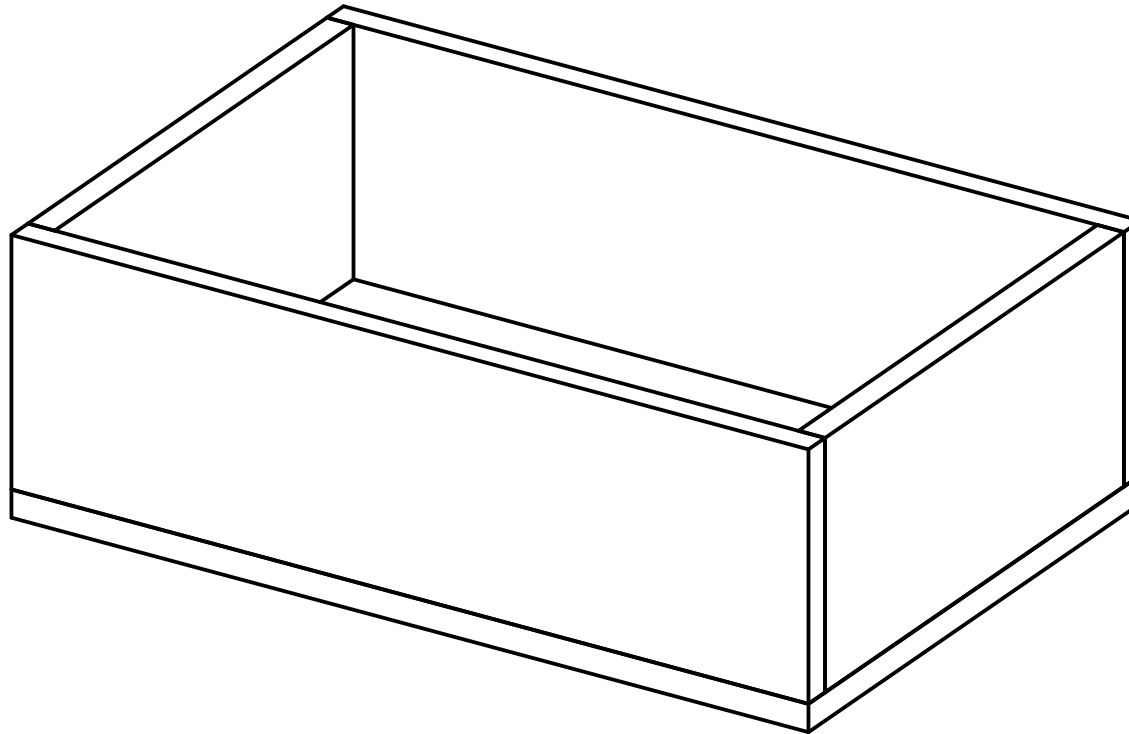


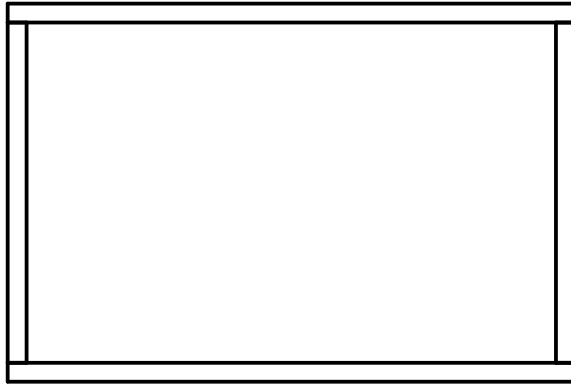
Butt box



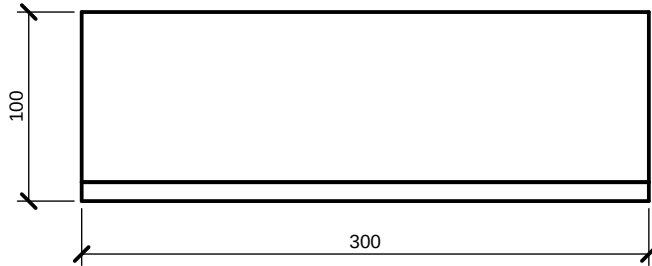
parts	5 (3 different)
overall	300 x 200 x 100 (W x D x H)
material	oak
date	2026-10-25
made with	woodwright 0.9.0

Contents

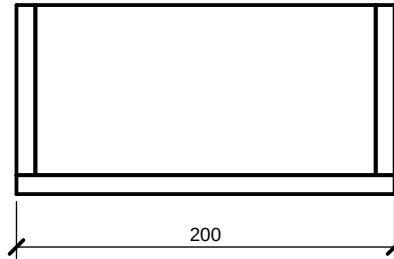
Drawings	3
Assembly	3
Parts	4
1 front_back	5
2 left_right	6
3 bottom	7
Cut list	8
Appendix: source	9



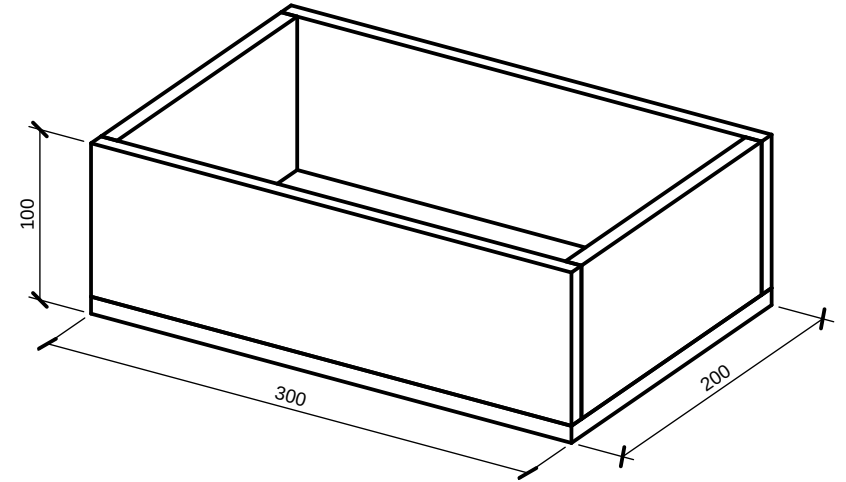
TOP



FRONT

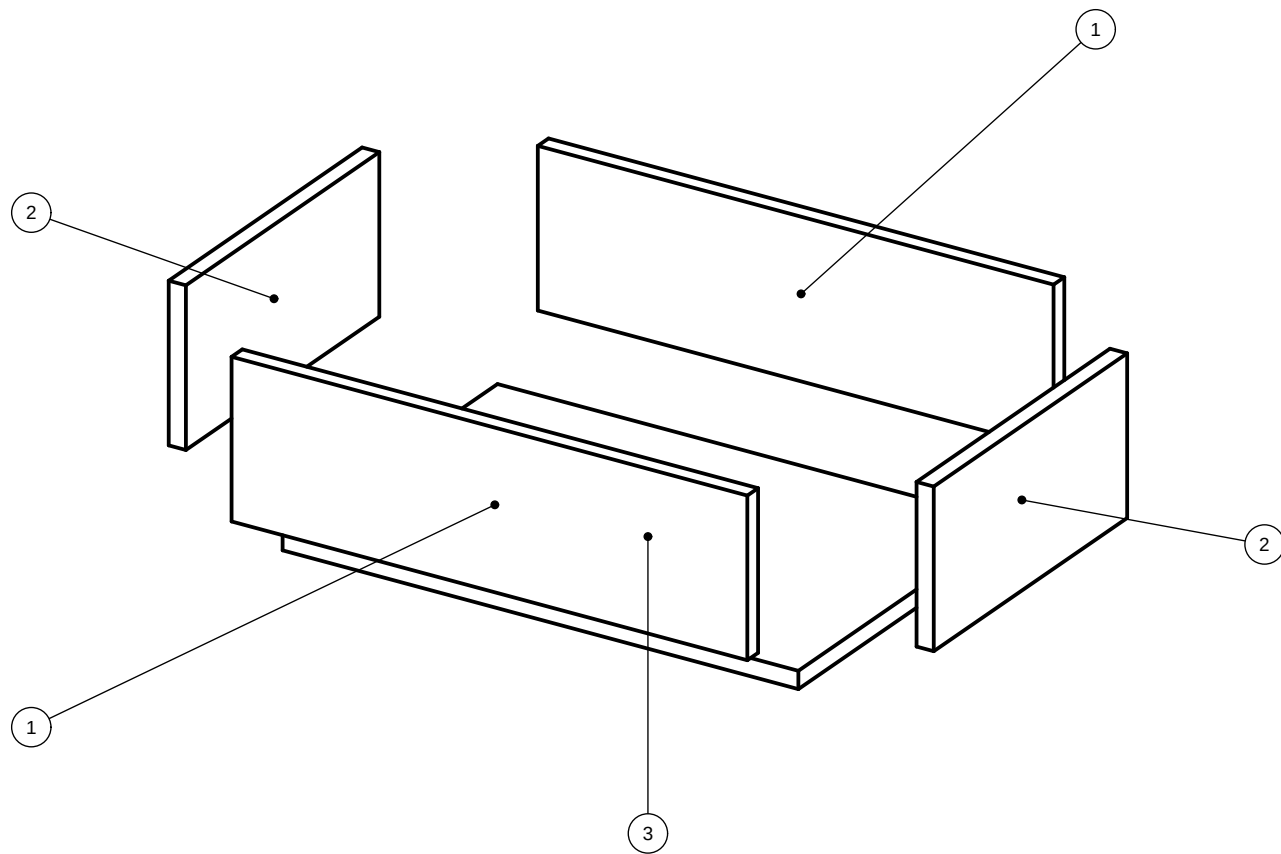


RIGHT SIDE



ISOMETRIC

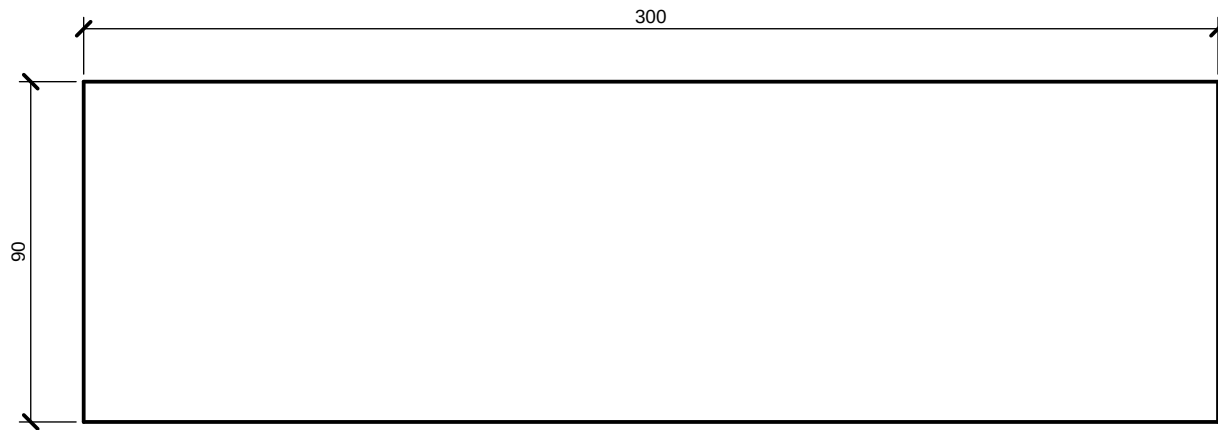
assembly	butt_box
parts	5 (3 different)
overall	300 x 200 x 100 (W x D x H)
material	oak
scale	1:4, mm



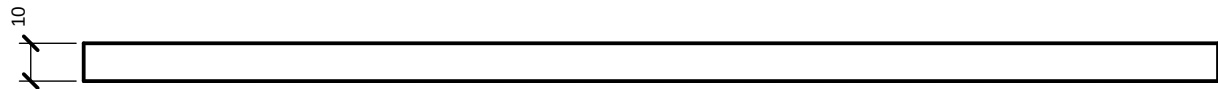
EXPLODED

no.	part	qty	cut size (L x W x T)	material
1	front_back	2	300 x 90 x 10	oak
2	left_right	2	180 x 90 x 10	oak
3	bottom	1	300 x 200 x 10	oak

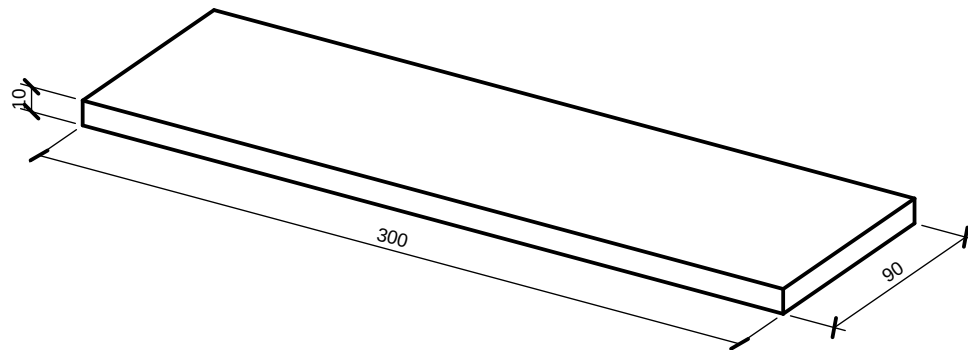
assembly	butt_box
parts	5 (3 different)
overall	300 x 200 x 100 (W x D x H)
material	oak
scale	not to scale, mm



TOP FACE



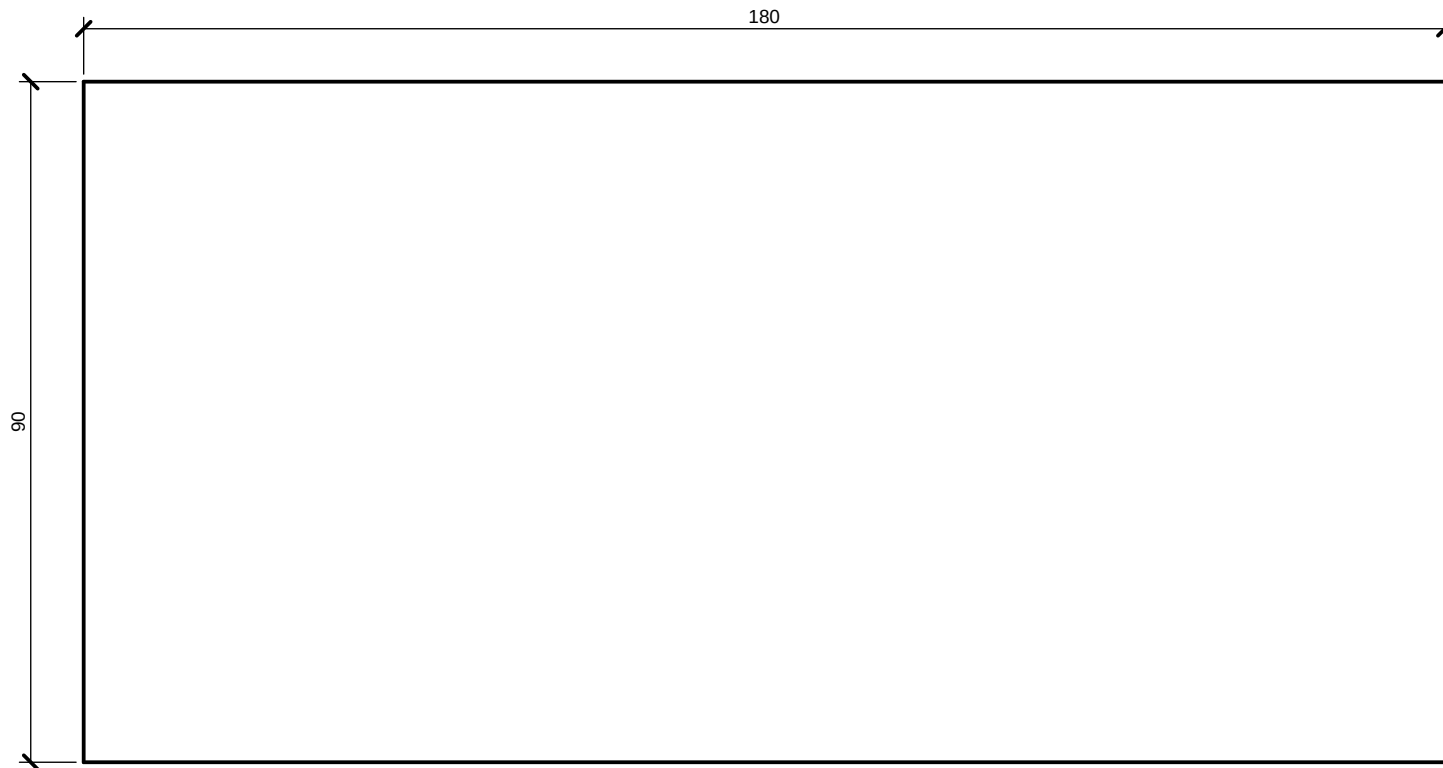
FRONT EDGE



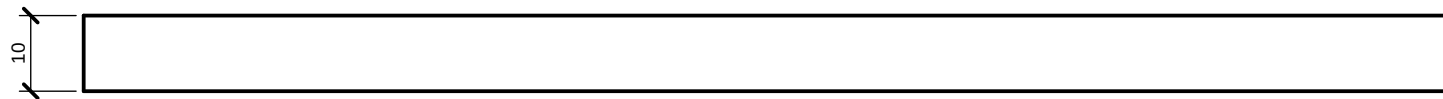
ISOMETRIC

2 pieces: front, back

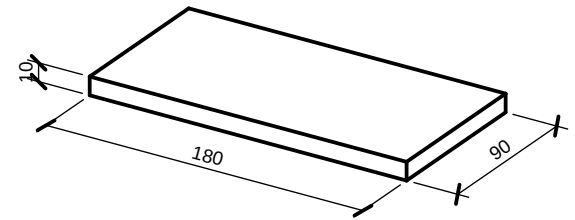
part	front_back
of	butt_box
make	2
cut size	300 x 90 x 10
material	oak
scale	1:2, mm



TOP FACE



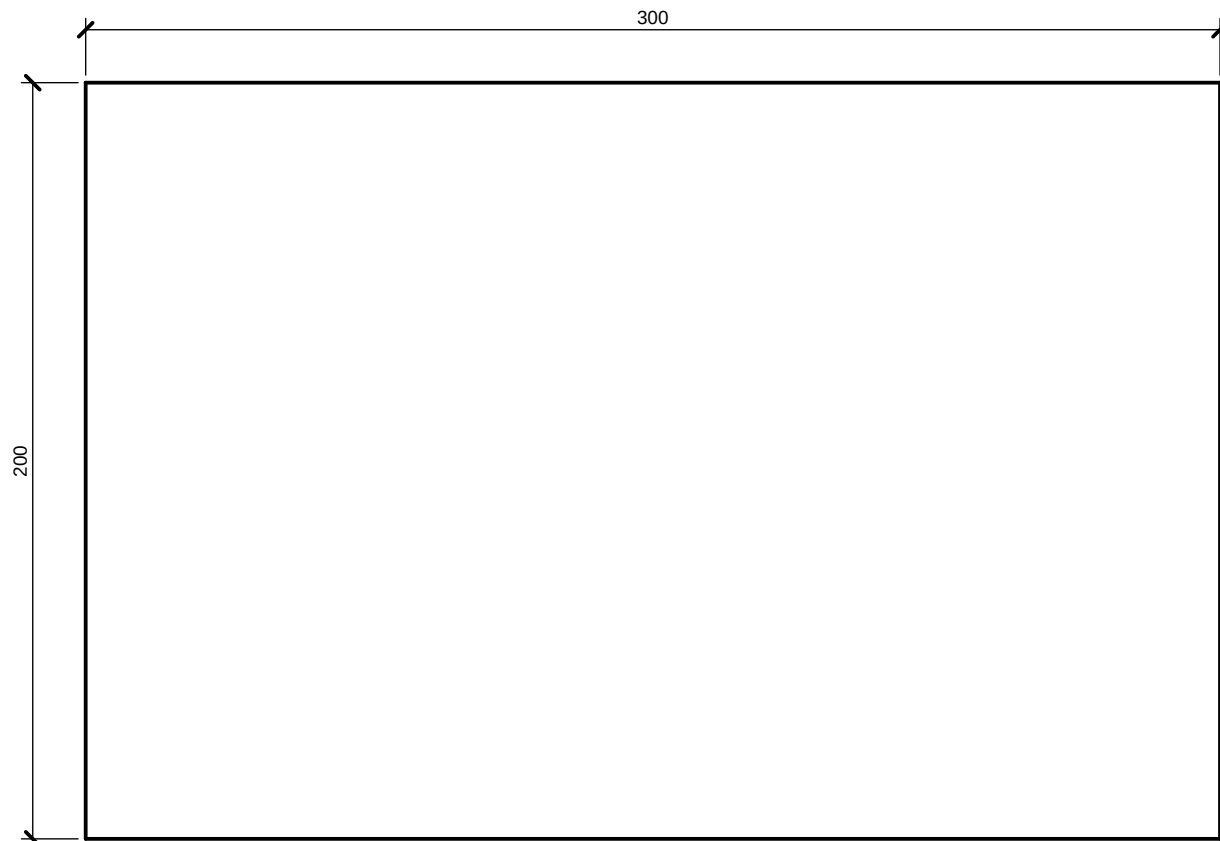
FRONT EDGE



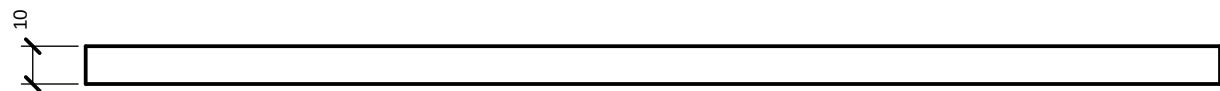
ISOMETRIC

2 pieces: left, right

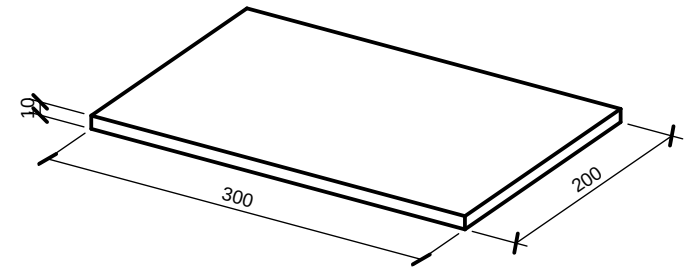
part	left_right
of	butt_box
make	2
cut size	180 x 90 x 10
material	oak
scale	1:1, mm



TOP FACE



FRONT EDGE



ISOMETRIC

part	bottom
of	butt_box
make	1
cut size	300 x 200 x 10
material	oak
scale	1:2, mm

Cut list

front_back	300 x 90 x 10 mm	oak, 2 pieces: front, back
left_right	180 x 90 x 10 mm	oak, 2 pieces: left, right
bottom	300 x 200 x 10 mm	oak

Appendix: source

Made with woodwright 0.9.0, build123d 0.13.0, Python 3.14.4.

To make these plans again:

```
uvx --python 3.14 woodwright==0.9.0 export projects.butt_box
```

projects.butt_box, in woodwright/examples/projects.py

```
def butt_box(
    length: float = 300,
    depth: float = 200,
    height: float = 100,
    stock: Stock = THIN_OAK,
) -> Assembly:
    """An open box: four walls butt-jointed at the corners on a bottom; outside sizes."""
    t = stock.thickness
    asm = Assembly("butt_box", root_location=BOX_FRONT)
    wall_height = height - t
    front = asm.add(Board("front", stock, length=length, width=wall_height))
    # The back and the right side are mirror images of the front and the
    # left side, all with their reference faces outside.
    back = asm.add(Board("back", stock, length=length, width=wall_height, mirror=True))
    left = asm.add(Board("left", stock, length=depth - 2 * t, width=wall_height))
    right = asm.add(Board("right", stock, length=depth - 2 * t, width=wall_height, mirror=True))
    bottom = asm.add(Board("bottom", stock, length=length, width=depth))
    # The sides between the front and the back, their bottom (front) edges level.
    for side, front_end, back_end in [
        (left, front.left_end, back.left_end),
        (right, front.right_end, back.right_end),
    ]:
        asm.join(Butt(front.bottom_face, side.left_end, position=flush_with(front_end)))
        asm.join(Butt(back.bottom_face, side.right_end, position=flush_with(back_end)))
    # The walls stand on the bottom, flush with its sides. The back faces
    # the other way than the front, so its reference face (outside) is flush
    # with the bottom's back edge.
    asm.join(Butt(front.front_edge, bottom.top_face, position=flush_with(front.left_end)))
    asm.join(
        Butt(
```

```
        back.front_edge,  
        bottom.top_face,  
        position=flush_with(back.left_end),  
        flush=(back.ref_face, bottom.back_edge),  
    )  
)  
return asm
```