Assembly Recipes

Simple Function Calls

```
@ <stdio.hdr>

fn bar()
{
    putchar('B');
}

fn foo()
{
    putchar('F');
    bar();
    putchar('f');
}

fn main()
{
    putchar('a');
    foo();
    putchar('b');
    foo();
    putchar('c');
    putchar('\n');
}
```

```
@ <stdio.hdr>

fn foo()
{
    putchar('F');
}

fn main()
{
    putchar('a');
    foo();
    putchar('b');
    putchar('\n');
}
```

```
@ <stdio.hdr>

fn foo(); // forward declaration

fn main()
{
    putchar('a');
    foo();
    putchar('b');
    putchar('\n');
}

fn foo()
{
    putchar('F');
}
```

```
@ <stdio.hdr>

fn foo(); // forward declaration

fn main()
{
    putchar('a');
    foo();
    putchar('b');
    putchar('\n');
}

fn foo()
{
    putchar('F');
}
```

```
    putc    'a'
    jmp     foo
ret putc    'b'
    putc    '\n'
    halt    0

foo putc    'F'
    jmp     ret
```

```
@ <stdio.hdr>

fn foo(); // forward declaration

fn main()
{
    putchar('a');
    foo();
    putchar('b');
    foo();
    putchar('c');
    putchar('\n');
}

fn foo()
{
    putchar('F');
}
```

```
@ <stdio.hdr>

fn foo(); // forward declaration

fn main()
{
    putchar('a');
    foo();
    putchar('b');
    foo();
    putchar('c');
    putchar('\n');
}

fn foo()
{
    putchar('F');
}
```

```
    putc    'a'
    jmp     foo
ret putc    'b'
    jmp     foo
ret putc    'c'         # error: labels have to be unique
    putc    '\n'
    halt    0


foo putc    'F'
    jmp     ret
```

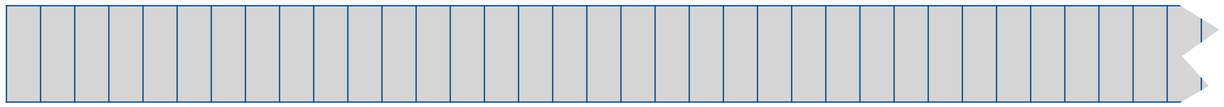
```
@ <stdio.hdr>

fn foo(); // forward declaration

fn main()
{
    putchar('a');
    foo();
    putchar('b');
    foo();
    putchar('c');
    putchar('\n');
}

fn foo()
{
    putchar('F');
}
```

```
@ <stdio.hdr>

fn foo(); // forward declaration

fn main()
{
    putchar('a');
    foo();
    putchar('b');
    foo();
    putchar('c');
    putchar('\n');
}

fn foo()
{
    putchar('F');
}
```

```
        putc    'a'
        call    foo,        %1
        putc    'b'
        call    foo,        %1
        putc    'c'
        putc    '\n'
        halt    0


foo     putc    'F'
        ret     %1
```

```
@ <stdio.hdr>

fn foo(); // forward declaration
fn bar(); // forward declaration

fn main()
{
    putchar('a');
    foo();
    putchar('b');
    foo();
    putchar('c');
    putchar('\n');
}

fn foo()
{
    putchar('F');
    bar();
    putchar('f');
}

fn bar()
{
    putchar('B');
}
```

```
        putc    'a'
        call    foo,        %1
        putc    'b'
        call    foo,        %1
        putc    'c'
        putc    '\n'
        halt    0


foo     putc    'F'
        call    bar,        %1 # will change %1 !!
        putc    'f'
        ret     %1


bar     putc    'B'
        ret     %1
```



```
@ <stdio.hdr>

fn foo(); // forward declaration
fn bar(); // forward declaration

fn main()
{
    putchar('a');
    foo();
    putchar('b');
    foo();
    putchar('c');
    putchar('\n');
}

fn foo()
{
    putchar('F');
    bar();
    putchar('f');
}

fn bar()
{
    putchar('B');
}
```

```
        .equ    SP,         1
        loadz   0,          %SP

main    putc    'a'
        call    foo,        %2
        putc    'b'
        call    foo,        %2
        putc    'c'
        putc    '\n'
        halt    0

foo     subq    8,          %SP,        %SP
        movq    %2,         (%SP)

        putc    'F'
        call    bar,        %2
        putc    'f'

        movq    (%SP),      %2
        addq    8,          %SP,        %SP
        ret     %2

bar     subq    8,          %SP,        %SP
        movq    %2,         (%SP)

        putc    'B'

        movq    (%SP),      %2
        addq    8,          %SP,        %SP
        ret     %2
```

```
        .equ    SP,         1
        loadz   0,          %SP

main    putc    'a'
        call    foo,        %2
        putc    'b'
        call    foo,        %2
        putc    'c'
        putc    '\n'
        halt    0

foo     subq    8,          %SP,        %SP
        movq    %2,         (%SP)

        putc    'F'
        call    bar,        %2
        putc    'f'

        movq    (%SP),      %2
        addq    8,          %SP,        %SP
        ret     %2

bar     subq    8,          %SP,        %SP
        movq    %2,         (%SP)

        putc    'B'

        movq    (%SP),      %2
        addq    8,          %SP,        %SP
        ret     %2
```

```
        .equ    SP,         1
        loadz   0,          %SP
        call    main,       %2
        halt    0

        # ...

main    subq    8,          %SP,        %SP
        movq    %2,         (%SP)

        putc    'a'
        call    foo,        %2
        putc    'b'
        call    foo,        %2
        putc    'c'
        putc    '\n'

        movq    (%SP),      %2
        addq    8,          %SP,        %SP
        ret     %2

        # ...
```

```
@ <stdio.hdr>

fn bar()
{
    putchar('B');
}

fn foo()
{
    putchar('F');
    bar();
    putchar('f');
}

fn main()
{
    putchar('a');
    foo();
    putchar('b');
    foo();
    putchar('c');
    putchar('\n');
}
```

```
        .equ    SP,         1
        loadz   0,          %SP
        call    main,       %2
        halt    0

bar     subq    8,          %SP,        %SP
        movq    %2,         (%SP)

        putc    'B'

        movq    (%SP),      %2
        addq    8,          %SP,        %SP
        ret     %2

foo     subq    8,          %SP,        %SP
        movq    %2,         (%SP)

        putc    'F'
        call    bar,        %2
        putc    'f'

        movq    (%SP),      %2
        addq    8,          %SP,        %SP
        ret     %2

main    subq    8,          %SP,        %SP
        movq    %2,         (%SP)

        putc    'a'
        call    foo,        %2
        putc    'b'
        call    foo,        %2
        putc    'c'
        putc    '\n'

        movq    (%SP),      %2
        addq    8,          %SP,        %SP
        ret     %2
```