

1. Desenhe o estado da pilha de execução após a execução da última instrução:

```
void main( ) {
    int x = 10;
    int *p_x = &x;
    int **p_p_x = &p_x;
    int ***p_p_p_x = &p_p_x;
    **p_p_p_x += 5;
    x+=10;
    *p_x = sqrt(*p_x);
}
```

p_p_p_x	0x0001	0x0004
p_p_x	0x0001	0x0003
p_x	0x0001	0x0002
x	10	0x0001
Soma 5 no espaço de memória 0x0001 via p_p_x		
p_p_p_x	0x0001	0x0004
p_p_x	0x0001	0x0003
p_x	0x0001	0x0002
x	15	0x0001
Soma 10 na variável x na memória 0x0001		

p_p_p_x	0x0001	0x0004
p_p_x	0x0001	0x0003
p_x	0x0001	0x0002
x	25	0x0001
Calcula raiz do valor na memória 0x0001 via p_x		
p_p_p_x	0x0001	0x0004
p_p_x	0x0001	0x0003
p_x	0x0001	0x0002
x	5	0x0001

2. Desenhe o estado da pilha de execução após a execução da última instrução:

```
void main( ){
    int *a, *b, c = 8, d = 2;
    a = &c;
    b = &d;
    *a = 5;
    (*a)+=10;
    *b = *a;
}
```

a	0x0002	0x0004
b	0x0001	0x0003
c	8	0x0002
d	2	0x0001
guarda 5 na memória 0x0002 via a		
a	0x0002	0x0004
b	0x0001	0x0003
c	5	0x0002
d	2	0x0001
Soma 10 no valor guardado na memória 0x0002		

a	0x0002	0x0004
b	0x0001	0x0003
c	15	0x0002
d	2	0x0001
Iguala os valores dentro das memórias apontadas		
a	0x0002	0x0004
b	0x0001	0x0003
c	15	0x0002
d	15	0x0001