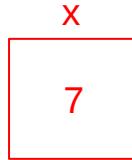


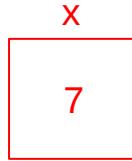
```
int x = 7;
if( fork() ) {
    x++;
    printf(" %d ", x);
    fork();
    x++;
    printf(" %d ", x);
} else {
    printf(" %d ", x);
}
```

```
int x = 7;  
if( fork() ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```



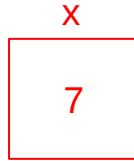
Este bloco de código é executado originalmente como um processo único, que podemos chamar de **processo vermelho**.

```
int x = 7;  
if( fork() ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```



Na primeira linha a variável local **x** está com o valor **7**

```
int x = 7;  
if( fork() ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```



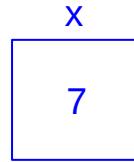
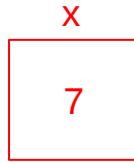
Nessa linha é executado o fork()

```
int x = 7;  
if( fork() ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

```
int x = 7;  
if( fork() ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```



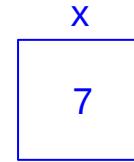
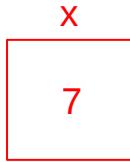
fork()



... e o processo é clonado

```
int x = 7;  
if( PID ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

```
int x = 7;  
if( fork() ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

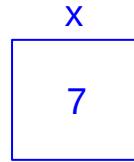
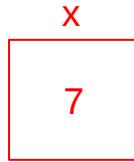


Observe que o estado dos dois processos são **idênticos**,  
exceto o fork() no processo vermelho que tem como  
retorno o **PID em azul**.

```
int x = 7;  
if( PID ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

```
int x = 7;  
if( 0 ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

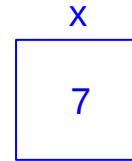
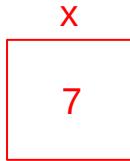
fork()



Enquanto que o fork() no processo azul retorna 0.

```
int x = 7;  
if( PID ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

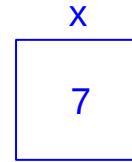
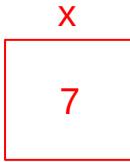
```
int x = 7;  
if( 0 ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```



Agora, o SO pode continuar executando tanto o processo vermelho ou processo azul.

```
int x = 7;  
if( PID ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

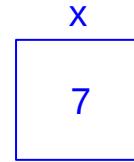
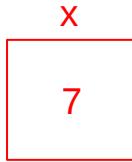
```
int x = 7;  
if( 0 ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```



Como existem muitos casos diferentes,  
vamos explorar apenas um dos casos.

```
int x = 7;  
if( PID ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

```
int x = 7;  
if( 0 ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

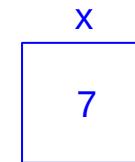
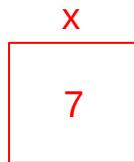


Considere que o **processo vermelho** continuará executando

```
int x = 7;  
if( PID ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```



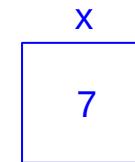
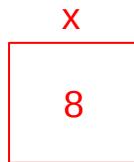
```
int x = 7;  
if( 0 ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```



Executando: processo vermelho...

```
int x = 7;  
if( PID ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

```
int x = 7;  
if( 0 ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

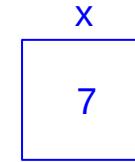
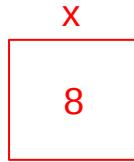


Executando: processo vermelho...

```
int x = 7;  
if( PID ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```



```
int x = 7;  
if( 0 ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```



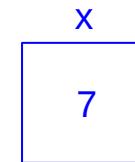
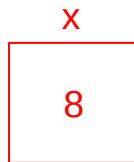
Output

8

Executando: processo vermelho...

```
int x = 7;  
if( PID ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

```
int x = 7;  
if( 0 ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```



Output

8

Executando: processo vermelho...

```
int x = 7;  
if( fork() ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

```
int x = 7;  
if( PID ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

```
int x = 7;  
if( 0 ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

x  
8

x  
8

x  
7

Output

8

Fork!

```
int x = 7;  
if( fork() ) {  
    x++;  
    printf(" %d ", x);  
    0 ;  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

```
int x = 7;  
if( PID ) {  
    x++;  
    printf(" %d ", x);  
    PID ;  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

```
int x = 7;  
if( 0 ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

x  
8

x  
8

x  
7

Output

8

Fork() retorna 0 para processo roxo e PID para processo vermelho

```
int x = 7;  
if( fork() ) {  
    x++;  
    printf(" %d ", x);  
    0 ;  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

```
int x = 7;  
if( PID ) {  
    x++;  
    printf(" %d ", x);  
    PID ;  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

```
int x = 7;  
if( 0 ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

x  
8

x  
8

x  
7

Output

8

Suponha que continuemos executando o processo vermelho.

```
int x = 7;  
if( fork() ) {  
    x++;  
    printf(" %d ", x);  
    0 ;  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

```
int x = 7;  
if( PID ) {  
    x++;  
    printf(" %d ", x);  
    PID ;  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

```
int x = 7;  
if( 0 ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

X  
8

X  
9

X  
7

Output

8

Executando: processo vermelho...

```
int x = 7;  
if( fork() ) {  
    x++;  
    printf(" %d ", x);  
    0 ;  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

```
int x = 7;  
if( PID ) {  
    x++;  
    printf(" %d ", x);  
    PID ;  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

```
int x = 7;  
if( 0 ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

x  
8

x  
9

x  
7

Output

8 9

Executando: processo vermelho...

```
int x = 7;  
if( fork() ) {  
    x++;  
    printf(" %d ", x);  
    0 ;  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

```
int x = 7;  
if( PID ) {  
    x++;  
    printf(" %d ", x);  
    PID ;  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

```
int x = 7;  
if( 0 ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

x  
8

x  
9

x  
7

Output

8 9

Executando: processo vermelho... Fim !!

```
int x = 7;  
if( fork() ) {  
    x++;  
    printf(" %d ", x);  
    0 ;  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

```
int x = 7;  
if( PID ) {  
    x++;  
    printf(" %d ", x);  
    PID ;  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

```
int x = 7;  
if( 0 ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

x  
8

x  
9

x  
7

Output

8 9

Agora executando o processo roxo...

```
int x = 7;  
if( fork() ) {  
    x++;  
    printf(" %d ", x);  
    0 ;  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

```
int x = 7;  
if( PID ) {  
    x++;  
    printf(" %d ", x);  
    PID ;  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

```
int x = 7;  
if( 0 ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

x  
9

x  
9

x  
7

Output

8 9

Executando: processo roxo...

```
int x = 7;  
if( fork() ) {  
    x++;  
    printf(" %d ", x);  
    0 ;  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

```
int x = 7;  
if( PID ) {  
    x++;  
    printf(" %d ", x);  
    PID ;  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

```
int x = 7;  
if( 0 ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

Fim!

x  
9

x  
9

x  
7

Output  
8 9 9

Executando: processo roxo...

```
int x = 7;  
if( fork() ) {  
    x++;  
    printf(" %d ", x);  
    0 ;  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

Fim!

x  
9

```
int x = 7;  
if( PID ) {  
    x++;  
    printf(" %d ", x);  
    PID ;  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

Fim!

x  
9

```
int x = 7;  
if( 0 ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```



Output  
8 9 9

x  
7

Executando: processo roxo...Fim!

```
int x = 7;  
if( fork() ) {  
    x++;  
    printf(" %d ", x);  
    0 ;  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

Fim!

x  
9

```
int x = 7;  
if( PID ) {  
    x++;  
    printf(" %d ", x);  
    PID ;  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

Fim!

x  
9

```
int x = 7;  
if( 0 ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```



x  
7

Output  
8 9 9

Finish executing the blue process.

```
int x = 7;  
if( fork() ) {  
    x++;  
    printf(" %d ", x);  
    0 ;  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

Fim!

x  
9

```
int x = 7;  
if( PID ) {  
    x++;  
    printf(" %d ", x);  
    PID ;  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

Fim!

x  
9

```
int x = 7;  
if( 0 ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```



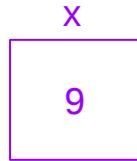
x  
7

Output  
8 9 9 7

Executando: processo azul...

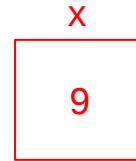
```
int x = 7;  
if( fork() ) {  
    x++;  
    printf(" %d ", x);  
    0 ;  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

Fim!



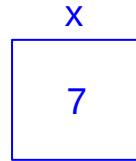
```
int x = 7;  
if( PID ) {  
    x++;  
    printf(" %d ", x);  
    PID ;  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

Fim!



```
int x = 7;  
if( 0 ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

Fim!

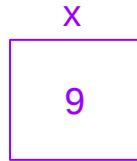


Output  
8 9 9 7

Executando: processo azul... Fim!

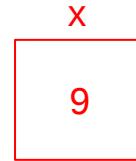
```
int x = 7;  
if( fork() ) {  
    x++;  
    printf(" %d ", x);  
    0 ;  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

Fim!



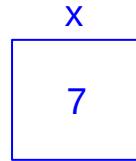
```
int x = 7;  
if( PID ) {  
    x++;  
    printf(" %d ", x);  
    PID ;  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

Fim!



```
int x = 7;  
if( 0 ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

Fim!

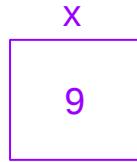


Output  
8 9 9 7

Então uma saída (Output) possível é “ 8 9 9 7 ”

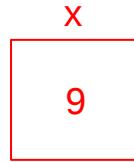
```
int x = 7;  
if( fork() ) {  
    x++;  
    printf(" %d ", x);  
    0 ;  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

Fim!



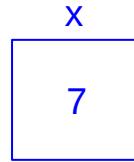
```
int x = 7;  
if( PID ) {  
    x++;  
    printf(" %d ", x);  
    PID ;  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

Fim!



```
int x = 7;  
if( 0 ) {  
    x++;  
    printf(" %d ", x);  
    fork();  
    x++;  
    printf(" %d ", x);  
} else {  
    printf(" %d ", x);  
}
```

Fim!



Output  
8 9 9 7

Tente encontrar outras saídas !