Tracing While Loops

Consider...

```
int i = 4, j = 1, n = 0;
while (i > j) {
    if (n % 2 == 0) {
        i--;
    } else {
        j++;
    }
    n++;
}
```

Trace It

`int i = 4, j = 1, n = 0;`	
	i = 4 j = 1 n = 0
`while (i > j) {`	
`  if (n % 2 == 0) {`	
`    i--;`	
`  } else {`	
`    j++;`	
`  }`	
`  n++;`	
`}`	

Trace It

`int i = 4, j = 1, n = 0;`	
	i = 4 j = 1 n = 0
`while (i > j) {`	
	i = 4 j = 1 n = 0
`  if (n % 2 == 0) {`	
`    i--;`	
`  } else {`	
`    j++;`	
`  }`	
`  n++;`	
`}`	

Trace It

`int i = 4, j = 1, n = 0;`	
	i = 4 j = 1 n = 0
`while (i > j) {`	
	i = 4 j = 1 n = 0
`  if (n % 2 == 0) {`	
	i = 4 j = 1 n = 0
`    i--;`	
`  } else {`	
`    j++;`	
`  }`	
`  n++;`	
`}`	

Trace It

`int i = 4, j = 1, n = 0;`	
	i = 4 j = 1 n = 0
`while (i > j) {`	
	i = 4 j = 1 n = 0
`  if (n % 2 == 0) {`	
	i = 4 j = 1 n = 0
`    i--;`	
	i = 3 j = 1 n = 0
`  } else {`	
`    j++;`	
`  }`	
`  n++;`	
`}`	

Trace It

`int i = 4, j = 1, n = 0;`	
	i = 4 j = 1 n = 0
`while (i > j) {`	
	i = 4 j = 1 n = 0
`  if (n % 2 == 0) {`	
	i = 4 j = 1 n = 0
`    i--;`	
	i = 3 j = 1 n = 0
`  } else {`	
	i = — j = — n = —
`    j++;`	
	i = — j = — n = —
`  }`	
	i = 3 j = 1 n = 0
`  n++;`	
`}`	

Trace It

`int i = 4, j = 1, n = 0;`	
	i = 4 j = 1 n = 0
`while (i > j) {`	
	i = 4 j = 1 n = 0
`  if (n % 2 == 0) {`	
	i = 4 j = 1 n = 0
`    i--;`	
	i = 3 j = 1 n = 0
`  } else {`	
	i = — j = — n = —
`    j++;`	
	i = — j = — n = —
`  }`	
	i = 3 j = 1 n = 0
`  n++;`	
	i = 3 j = 1 n = 1
`}`	

Trace It

`int i = 4, j = 1, n = 0;`	
	i = 4 j = 1 n = 0
`while (i > j) {`	
	i = ~~4~~ 3 j = ~~1~~ 1 n = ~~0~~ 1
`  if (n % 2 == 0) {`	
	i = 4 j = 1 n = 0
`    i--;`	
	i = 3 j = 1 n = 0
`  } else {`	
	i = — j = — n = —
`    j++;`	
	i = — j = — n = —
`  }`	
	i = 3 j = 1 n = 0
`  n++;`	
	i = 3 j = 1 n = 1
`}`	

Trace It

`int i = 4, j = 1, n = 0;`	
	i = 4 j = 1 n = 0
`while (i > j) {`	
	i = ~~4~~ 3 j = ~~1~~ 1 n = ~~0~~ 1
`  if (n % 2 == 0) {`	
	i = ~~4~~ — j = ~~1~~ — n = ~~0~~ —
`    i--;`	
	i = ~~3~~ — j = ~~1~~ — n = ~~0~~ —
`  } else {`	
	i = — 3 j = — 1 n = — 1
`    j++;`	
	i = — j = — n = —
`  }`	
	i = 3 j = 1 n = 0
`  n++;`	
	i = 3 j = 1 n = 1
`}`	

Trace It

`int i = 4, j = 1, n = 0;`			
	i = 4	*j = 1*	*n = 0*
`while (i > j) {`			
	i = ~~4~~ 3	*j = ~~1~~ 1*	*n = ~~0~~ 1*
`  if (n % 2 == 0) {`			
	i = ~~4~~ —	*j = ~~1~~ —*	*n = ~~0~~ —*
`    i--;`			
	i = ~~3~~ —	*j = ~~1~~ —*	*n = ~~0~~ —*
`  } else {`			
	i = — 3	*j = — 1*	*n = — 1*
`    j++;`			
	i = — 3	*j = — 2*	*n = — 1*
`  }`			
	i = 3	*j = 1*	*n = 0*
`  n++;`			
	i = 3	*j = 1*	*n = 1*
`}`			

Trace It

`int i = 4, j = 1, n = 0;`	
	i = 4 j = 1 n = 0
`while (i > j) {`	
	i = ~~4~~ 3 j = ~~1~~ 1 n = ~~0~~ 1
`  if (n % 2 == 0) {`	
	i = ~~4~~ — j = ~~1~~ — n = ~~0~~ —
`    i--;`	
	i = ~~3~~ — j = ~~1~~ — n = ~~0~~ —
`  } else {`	
	i = — 3 j = — 1 n = — 1
`    j++;`	
	i = — 3 j = — 2 n = — 1
`  }`	
	i = ~~3~~ 3 j = ~~1~~ 2 n = ~~0~~ 1
`  n++;`	
	i = 3 j = 1 n = 1
`}`	

Trace It

`int i = 4, j = 1, n = 0;`			
	i = 4	*j = 1*	*n = 0*
`while (i > j) {`			
	i = ~~4~~ 3	*j = ~~1~~ 1*	*n = ~~0~~ 1*
`  if (n % 2 == 0) {`			
	i = ~~4~~ —	*j = ~~1~~ —*	*n = ~~0~~ —*
`    i--;`			
	i = ~~3~~ —	*j = ~~1~~ —*	*n = ~~0~~ —*
`  } else {`			
	i = — 3	*j = — 1*	*n = — 1*
`    j++;`			
	i = — 3	*j = — 2*	*n = — 1*
`  }`			
	i = ~~3~~ 3	*j = ~~1~~ 2*	*n = ~~0~~ 1*
`  n++;`			
	i = ~~3~~ 3	*j = ~~1~~ 2*	*n = ~~1~~ 2*
`}`			

Trace It

`int i = 4, j = 1, n = 0;`			
	i = 4	*j = 1*	*n = 0*
`while (i > j) {`			
	i = ~~4~~ ~~3~~ 3	*j = ~~1~~ ~~1~~ 2*	*n = ~~0~~ ~~1~~ 2*
`  if (n % 2 == 0) {`			
	i = ~~4~~ —	*j = ~~1~~ —*	*n = ~~0~~ —*
`    i--;`			
	i = ~~3~~ —	*j = ~~1~~ —*	*n = ~~0~~ —*
`  } else {`			
	i = — 3	*j = — 1*	*n = — 1*
`    j++;`			
	i = — 3	*j = — 2*	*n = — 1*
`  }`			
	i = ~~3~~ 3	*j = ~~1~~ 2*	*n = ~~0~~ 1*
`  n++;`			
	i = ~~3~~ 3	*j = ~~1~~ 2*	*n = ~~1~~ 2*
`}`			

Trace It

`int i = 4, j = 1, n = 0;`			
	i = 4	*j = 1*	*n = 0*
`while (i > j) {`			
	i = ~~4~~ ~~3~~ 3	*j = ~~1~~ ~~1~~ 2*	*n = ~~0~~ ~~1~~ 2*
`  if (n % 2 == 0) {`			
	i = ~~4~~ – 3	*j = ~~1~~ – 2*	*n = ~~0~~ – 2*
`    i--;`			
	i = ~~3~~ –	*j = ~~1~~ –*	*n = ~~0~~ –*
`  } else {`			
	i = – 3	*j = – 1*	*n = – 1*
`    j++;`			
	i = – 3	*j = – 2*	*n = – 1*
`  }`			
	i = ~~3~~ 3	*j = ~~1~~ 2*	*n = ~~0~~ 1*
`  n++;`			
	i = ~~3~~ 3	*j = ~~1~~ 2*	*n = ~~1~~ 2*
`}`			

Trace It

`int i = 4, j = 1, n = 0;`	
	i = 4 j = 1 n = 0
`while (i > j) {`	
	i = ~~4~~ ~~3~~ 3 j = ~~1~~ ~~1~~ 2 n = ~~0~~ ~~1~~ 2
`  if (n % 2 == 0) {`	
	i = ~~4~~ – 3 j = ~~1~~ – 2 n = ~~0~~ – 2
`    i--;`	
	i = ~~3~~ – 2 j = ~~1~~ – 2 n = ~~0~~ – 2
`  } else {`	
	i = – 3 j = – 1 n = – 1
`    j++;`	
	i = – 3 j = – 2 n = – 1
`  }`	
	i = ~~3~~ 3 j = ~~1~~ 2 n = ~~0~~ 1
`  n++;`	
	i = ~~3~~ 3 j = ~~1~~ 2 n = ~~1~~ 2
`}`	

Trace It

`int i = 4, j = 1, n = 0;`	
	i = 4 j = 1 n = 0
`while (i > j) {`	
	i = ~~4~~ ~~3~~ 3 j = ~~1~~ ~~1~~ 2 n = ~~0~~ ~~1~~ 2
`  if (n % 2 == 0) {`	
	i = ~~4~~ – 3 j = ~~1~~ – 2 n = ~~0~~ – 2
`    i--;`	
	i = ~~3~~ – 2 j = ~~1~~ – 2 n = ~~0~~ – 2
`  } else {`	
	i = – ~~3~~ – j = – ~~1~~ – n = – ~~1~~ –
`    j++;`	
	i = – ~~3~~ – j = – ~~2~~ – n = – ~~1~~ –
`  }`	
	i = ~~3~~ ~~3~~ 2 j = ~~1~~ ~~2~~ 2 n = ~~0~~ ~~1~~ 2
`  n++;`	
	i = ~~3~~ 3 j = ~~1~~ 2 n = ~~1~~ 2
`}`	

Trace It

`int i = 4, j = 1, n = 0;`	
	i = 4 j = 1 n = 0
`while (i > j) {`	
	i = ~~4~~ ~~3~~ 3 j = ~~1~~ ~~1~~ 2 n = ~~0~~ ~~1~~ 2
`  if (n % 2 == 0) {`	
	i = ~~4~~ – 3 j = ~~1~~ – 2 n = ~~0~~ – 2
`    i--;`	
	i = ~~3~~ – 2 j = ~~1~~ – 2 n = ~~0~~ – 2
`  } else {`	
	i = – ~~3~~ – j = – ~~1~~ – n = – ~~1~~ –
`    j++;`	
	i = – ~~3~~ – j = – ~~2~~ – n = – ~~1~~ –
`  }`	
	i = ~~3~~ ~~3~~ 2 j = ~~1~~ ~~2~~ 2 n = ~~0~~ ~~1~~ 2
`  n++;`	
	i = ~~3~~ ~~3~~ 2 j = ~~1~~ ~~2~~ 2 n = ~~1~~ ~~2~~ 3
`}`	

Trace It

`int i = 4, j = 1, n = 0;`	
	i = 4 j = 1 n = 0
`while (i > j) {`	
	i = ~~4~~ ~~3~~ 3 j = ~~1~~ ~~1~~ 2 n = ~~0~~ ~~1~~ 2
`  if (n % 2 == 0) {`	
	i = ~~4~~ – 3 j = ~~1~~ – 2 n = ~~0~~ – 2
`    i--;`	
	i = ~~3~~ – 2 j = ~~1~~ – 2 n = ~~0~~ – 2
`  } else {`	
	i = – ~~3~~ – j = – ~~1~~ – n = – ~~1~~ –
`    j++;`	
	i = – ~~3~~ – j = – ~~2~~ – n = – ~~1~~ –
`  }`	
	i = ~~3~~ ~~3~~ 2 j = ~~1~~ ~~2~~ 2 n = ~~0~~ ~~1~~ 2
`  n++;`	
	i = ~~3~~ ~~3~~ 2 j = ~~1~~ ~~2~~ 2 n = ~~1~~ ~~2~~ 3
`}`	
	i = 2 j = 2 n = 3

What Does It Do?

- Describe in one short sentence what the snippet of code we just traced does.
- Don’t repeat the code in English…