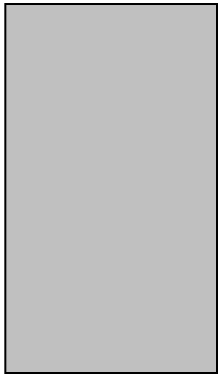
Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```



N = 6

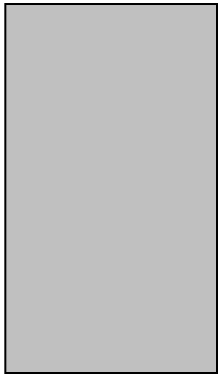
Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

i
0

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```



N = 6

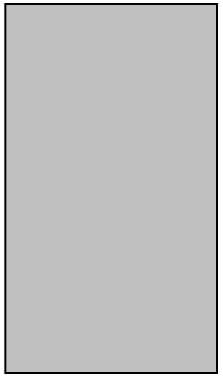
Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v
0	1



N = 6

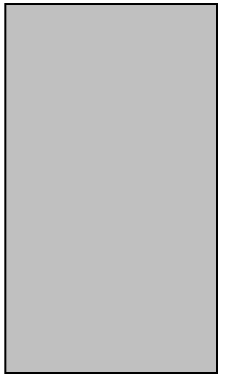
Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true



N = 6

Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true

```
0 1
```

N = 6

Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true
1		

```
0 1
```

N = 6

Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true
1	2	

```
0 1
```

N = 6

Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true
1	2	true

```
0 1
```

N = 6

Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true
1	2	true

```
0 1
1 2
```

N = 6

Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true
1	2	true
2		

```
0 1
1 2
```

N = 6

Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true
1	2	true
2	4	

```
0 1
1 2
```

N = 6

Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true
1	2	true
2	4	true

```
0 1
1 2
```

N = 6

Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true
1	2	true
2	4	true

```
0 1
1 2
2 4
```

N = 6

Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true
1	2	true
2	4	true
3		

```
0 1
1 2
2 4
```

N = 6

Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true
1	2	true
2	4	true
3	8	

```
0 1
1 2
2 4
```

N = 6

Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true
1	2	true
2	4	true
3	8	true

```
0 1
1 2
2 4
```

N = 6

Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true
1	2	true
2	4	true
3	8	true

```
0 1
1 2
2 4
3 8
```

N = 6

Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true
1	2	true
2	4	true
3	8	true
4		

```
0 1
1 2
2 4
3 8
```

N = 6

Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true
1	2	true
2	4	true
3	8	true
4	16	

```
0 1
1 2
2 4
3 8
```

N = 6

Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true
1	2	true
2	4	true
3	8	true
4	16	true

```
0 1
1 2
2 4
3 8
```

N = 6

Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true
1	2	true
2	4	true
3	8	true
4	16	true

```
0 1
1 2
2 4
3 8
4 16
```

N = 6

Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true
1	2	true
2	4	true
3	8	true
4	16	true
5		

```
0 1
1 2
2 4
3 8
4 16
```

N = 6

Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true
1	2	true
2	4	true
3	8	true
4	16	true
5	32	

```
0 1
1 2
2 4
3 8
4 16
```

N = 6

Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true
1	2	true
2	4	true
3	8	true
4	16	true
5	32	true

```
0 1
1 2
2 4
3 8
4 16
```

N = 6

Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true
1	2	true
2	4	true
3	8	true
4	16	true
5	32	true

```
0 1
1 2
2 4
3 8
4 16
5 32
```

N = 6

Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true
1	2	true
2	4	true
3	8	true
4	16	true
5	32	true
6		

```
0 1
1 2
2 4
3 8
4 16
5 32
```

N = 6

Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true
1	2	true
2	4	true
3	8	true
4	16	true
5	32	true
6	64	

```
0 1
1 2
2 4
3 8
4 16
5 32
```

N = 6

Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true
1	2	true
2	4	true
3	8	true
4	16	true
5	32	true
6	64	true

```
0 1
1 2
2 4
3 8
4 16
5 32
```

N = 6

Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true
1	2	true
2	4	true
3	8	true
4	16	true
5	32	true
6	64	true

```
0 1
1 2
2 4
3 8
4 16
5 32
6 64
```

N = 6

Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true
1	2	true
2	4	true
3	8	true
4	16	true
5	32	true
6	64	true
7		

```
0 1
1 2
2 4
3 8
4 16
5 32
6 64
```

N = 6

Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true
1	2	true
2	4	true
3	8	true
4	16	true
5	32	true
6	64	true
7	128	

N = 6

```
0 1
1 2
2 4
3 8
4 16
5 32
6 64
```

Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true
1	2	true
2	4	true
3	8	true
4	16	true
5	32	true
6	64	true
7	128	false

N = 6

```
0 1
1 2
2 4
3 8
4 16
5 32
6 64
```

Powers of Two: Trace

Ex. Print powers of 2 that are $\leq 2^N$

- Increment `i` from `0` to `N`
- Double `v` each time

```
int i = 0;
int v = 1;
while (i <= N) {
   System.out.println(i + " " + v);
   i = i + 1;
   v = 2 * v;
}
```

i	v	i <= N
0	1	true
1	2	true
2	4	true
3	8	true
4	16	true
5	32	true
6	64	true
7	128	false

```
0 1
1 2
2 4
3 8
4 16
5 32
6 64
```

N = 6