

CS 10

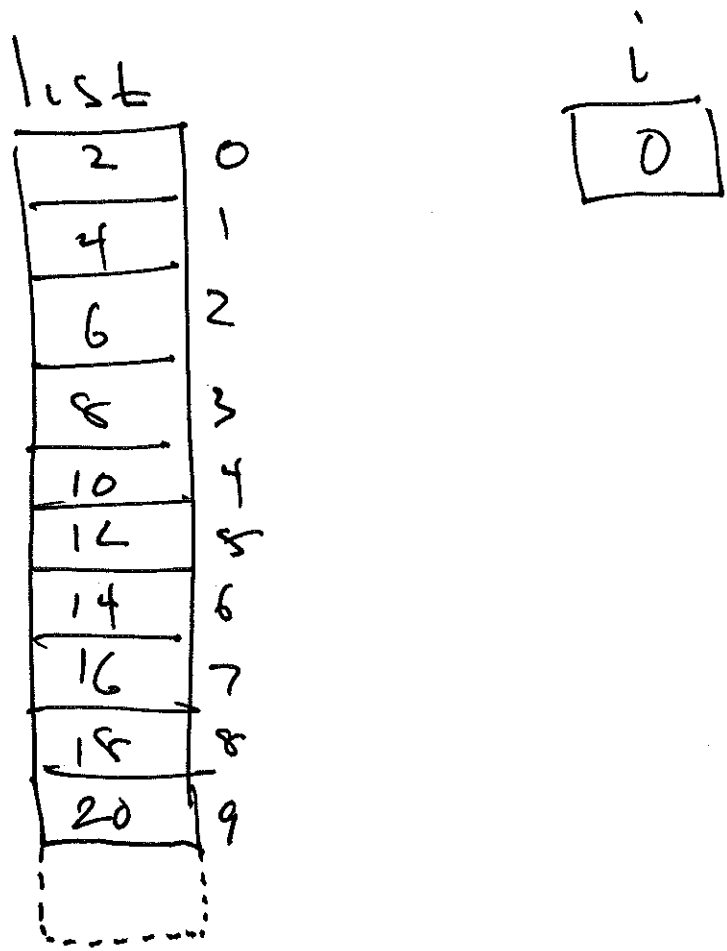
12-1-09



Ex.

```
int i;  
int list[10] = {2, 4, 6, 8, 10, 12, 14, 16, 18, 20};  
i = 0;  
while (i < 10) {  
    cout << list[i] << ' ';  
    i = i + 1;  
}  
cout << endl;
```

Picture of memory :



TRY

cout << list[10] << endl;

list[10] = 50;

Ex.

```
int i;  
int list[10] = {2, 4, 6, 8, 10, 12, 14, 16, 18, 20};  
i = 0;  
do {  
    cout << list[i] << " ";  
    i = i + 1;  
} while(i < 10);  
cout << endl;
```

```
for (initialize LCV; test LRC; increment LCV) {  
    :  
}
```

EX `int i, list[10] = {2, 4, 6, 8, 10, 12, 14, 16, 18, 20};`

```
for (i=0; i<10; i=i+1) {  
    cout << list[i] << " ";  
}  
cout << endl;
```

Compound Assignment OPS

Shortcut

$a = a + b;$

$a += b;$

$a = a - b;$

$a -= b;$

$a = a * b;$

$a *= b;$

$a = a / b;$

$a /= b;$

Ex.

```
for (i = 0; i < 10; i += 1) {  
    :  
}
```

AUTO-Increment / AUTO Decrement OPS

Shorthcuts

	<u>Postfix</u>	<u>Prefix</u>
$a += 1;$	$a++;$	$++a;$
$a -= 1;$	$a--;$	$--a;$

Ex. `for (i = 0; i < 10; i++) {`
`:`
`}`

Difference : Pre vs. Post.

Ex. `int a = 1, b;`
`b = a++;`
`cout << b << endl;`
`cout << a << endl;`



Ex.

```
int a = 1, b;
```

```
b = ++a;
```

```
cout << b << endl;
```

output:

2
2

Selection Sort (in: $n \geq 1, a_1, \dots, a_n$)

- 1.) $R \leftarrow n$
- 2.) while $R \geq 2$
- 3.) find index i of max element in unsorted section
- 4.) swap $a_i \leftrightarrow a_R$
- 5.) $R \leftarrow R - 1$
- 6.) stop

```

3.1)  $i \leftarrow 1$ 
3.2)  $j \leftarrow 2$ 
3.3) while  $j \leq R$ 
3.4)     if  $a_j > a_i$ 
3.5)          $i \leftarrow j$ 
3.6)      $j \leftarrow j + 1$ 
    
```

```

4.1) temp  $\leftarrow a_i$ 
4.2)  $a_i \leftarrow a_R$ 
4.3)  $a_R \leftarrow temp$ 
    
```

See SelectionSort.cpp on webpage

Bubble Sort ($n: n \geq 1, a_1, \dots, a_n$)

- 1.) $R \leftarrow n$
- 2.) while $R \geq 2$
- 3.) $j \leftarrow 2$
- 4.) while $j \leq R$
- 5.) if $a_j < a_{j-1}$
- 6.) swap $a_j \leftrightarrow a_{j-1}$
- 7.) $j \leftarrow j+1$
- 8.) $R \leftarrow R-1$
- 9.) stop

6.1) $temp \leftarrow a_j$
 6.2) $a_j \leftarrow a_{j-1}$
 6.3) $a_{j-1} \leftarrow temp$

See BubbleSort.cpp on
 webpage.

Insertion Sort ($n: n \geq 1, a_1, \dots, a_n$)

- 1.) $L \leftarrow 2$
- 2.) while $L \leq n$
- 3.) $j \leftarrow L$
- 4.) while $j \geq 2$ and $a_j < a_{j-1}$
- 5.) swap $a_j \leftrightarrow a_{j-1}$
- 6.) $j \leftarrow j - 1$
- 7.) $L \leftarrow L + 1$
- 8.) stop



Refine.....

Performance (# of comparisons)

	<u>Best</u>	<u>worst</u>	<u>avg.</u>
• Selection:	$\frac{1}{2}n^2 - \frac{1}{2}n$	$\frac{1}{2}n^2 - \frac{1}{2}n$	$\frac{1}{2}n^2 - \frac{1}{2}n$
• Bubble:	$\frac{1}{2}n^2 - \frac{1}{2}n$	$\frac{1}{2}n^2 - \frac{1}{2}n$	$\frac{1}{2}n^2 - \frac{1}{2}n$
• Insertion:	n	$\frac{1}{2}n^2 - \frac{1}{2}n$	$\frac{1}{4}n^2 + \frac{3}{4}n - \ln(n)$
• Quick:	$\Theta(n)$	$\Theta(n^2)$	$\begin{cases} 2n \ln(n) - 4n + 2 \ln(n) \\ \Theta(n \ln(n)) \end{cases}$
• Merge:	$\Theta(n)$	$\Theta(n \log n)$	$\Theta(n \log n)$