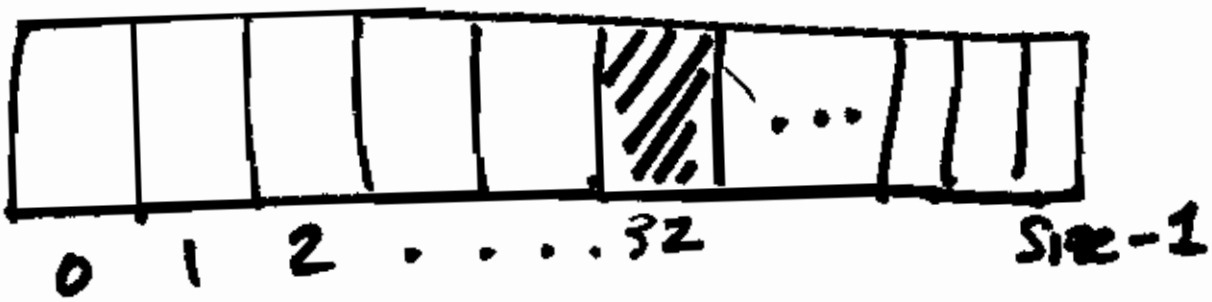
Client Interface

Function Overloading

```
class ordered_list {
  public:
     ordered_list();
     ~ordered_list();
     int insert(int, const data & );
     int retrieve(int, data &);
     int display();
     int remove(int);
     int retrieve (int start, int end,
                   data [], int &size, );
     int display (int start, int end);
```

Absolute List

Statically Allocated Array



✓ Direct Access

array[32] = ...

(*)(array (+) 32)

↓

32 (*) sizeof(data)

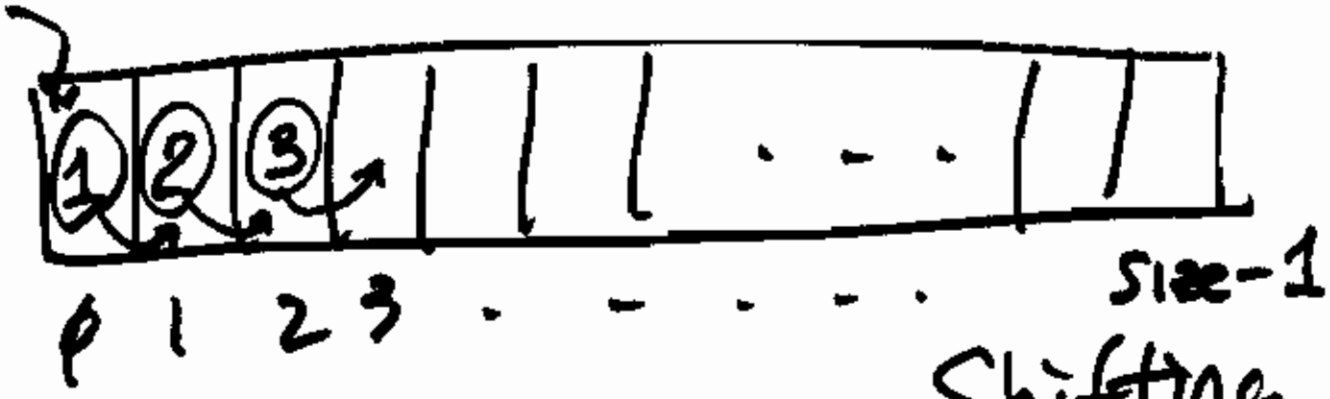
temporary

4 operations & 3 fetches ==> 7

→ X Memory - how much? Guess...
X " - Waste (sparse)
X " - Can you allocate it?

Relative List

Statically Allocated



X - 10,000 items Shifting

X Full !

Absolute List

Dynamically allocated Array

```
ordered_list::ordered_list(int num)
{  if (num <= 0) num = SIZE;
   array = new data[num];
   number_of_items = 0;
   size_of_array = num;
   // initialize the array of data
   for (int i = 0; i < num; ++i)
      array[i].setinthal();
}
```

can be skipped if data is a class with default constructor

Absolute List

LLL

✓ Memory!

Not contiguous
No Guess
No Waste.....

10,000 * addresses

```
current = head;
while (current != current->data < position)
{
        previous = current;
        current = current->next;
}
```

data-being-added (position)

7 fetches + 8 ops = ~15

15 * 9998

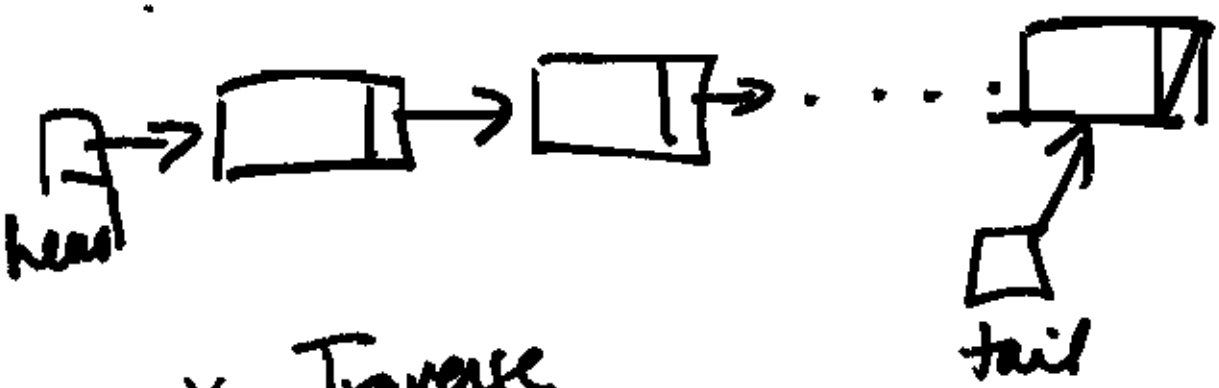
if add at end

Data to be added

```
if ( tail->data < position )
{
        tail->next = new node;
        tail = tail->next;
        tail->data = ....;
        tail->next = Null;
}
```

Relative Ordered List

LLL



✗ Traverse

✓ No Shift