Circular Linked List

```cpp
    "first"

    cout << ptr->next->data
    "first"

    if (List is Empty)
```
Display all LL

node * temp = head;

while ( temp )
{
    cout << temp->data << endl;
    temp = temp->next;
}

(Also Display all for DL)
Display-all CLL

1) Special case
   if (!ptr) return 

2) node * temp = ptr->next;
   do
      cout << temp->data
      cout << endl;
      temp = temp->next;
   while (temp != ptr->next);

~ (o op. & fetched)
Shortcut for CLL

```c
node * hold = ptr->next;
ptr->next = NULL;
temp = hold;
while (temp)
{
    cout << temp->data << endl;
    temp = temp->next;
}
ptr->next = hold;
```
Doubly Linked List

```
struct node
{
    student data;
    node * next;
    node * previous;
}
```

10,000 Nodes we have:
20,000 Addresses + 2
DLL Display Backward

node * temp = tail;
while (temp)
{
    cout << temp->data << endl;
    temp = temp->previous;
}

struct node
{
    student array[4];    // statically
    node * next;
};

struct node
{
    student * array;    // dynamically
    node * next;
};

SIZE = 1000

"absolute"

LLL of Arrays

1000 & students
4 students
1000

student
**First Node**

```c
if(!head)
    head = new node;
    head->array = new student [size];
    head->next = NULL;
```

- **Saving data**
  ```c
  head->array[i].copyStudent(&student);
  
  or
  
  head->array[index].copyStudent(...);
  ```

- **When adding**
  ```c
  if (head && index < size )
      head->array[index] = ...;
      ++index;
  
  else if ( index >= size )
      index = 0;
  ```
Go To Position 300 (Absolute)

however to = 300 / 512

which element = 300 % 512

current = next

current = array [index].copy...

current = next

current = next
Array of Linear Linked List

"Relahe"

100 nodes

NO Shifting!
what about?