

	Danh sách liên kết [F ₁]	Danh sách liên kết [F ₂]
Khởi báo Node	<pre>struct node { int data; node *next; };</pre>	<pre>struct node { int data; node *next; };</pre>
Khởi báo danh sách	<pre>struct LinkedList { node *head; };</pre>	
Khởi tạo node	<pre>node *createnode(int x) { node *temp = new node; temp->data = x; temp->next = NULL; return temp; }</pre>	<pre>node *createnode(int x) { node *temp = new node; temp->data = x; temp->next = NULL; return temp; }</pre>
Khởi tạo danh sách	<pre>void createlist(LinkedList &l) { l.head = NULL; }</pre>	
Chèn vào đầu danh sách	<pre>void addHead(LinkedList &l, int x) { if(l.head == NULL) { l.head = createnode(x); } else if(l.head != NULL) { node *temp = createnode(x); temp->next = l.head; l.head = temp; } }</pre>	<pre>node *addHead(node *l, int x) { if(l == NULL) { l = createnode(x); return l; } else if(l != NULL) { node *temp = createnode(x); temp->next = l; l = temp; return l; } }</pre>
Chèn vào cuối danh sách	<pre>void addTail(LinkedList &l, int x) { if(l.head == NULL) { l.head = createnode(x); } else { node *p = l.head; while(p->next != NULL) { p = p->next; } node *temp = createnode(x); p->next = temp; } }</pre>	<pre>node *addTail(node *l, int x) { if(l == NULL) { l = createnode(x); return l; } if(l != NULL) { node *p = l; while(p->next != NULL) { p = p->next; } node *temp = createnode(x); p->next = temp; return l; } }</pre>

Hàm nhập danh sách	<pre> void Ip_list(LinkedList &list) { createlist(list); int n,x; cin>>n; for(int i = 0 ; i < n; i++) { cin>>x; addTail(list,x); } } </pre>	<pre> node *Ip_List(node *l) { int x,n; cin>>n; for(int i = 0 ; i < n; i++) { cin>>x; l = addTail(l,x); } return l; } </pre>
Hàm in ra danh sách	<pre> void print_List(LinkedList l) { node *p = l.head; while(p != NULL) { cout<<p->data<<" "; p = p->next; } } </pre>	<pre> void print_List(node *l) { node *p = l; while(p != NULL) { cout<<p->data<<" "; p = p->next; } } </pre>
Trong hàm main()	<pre> int main() { LinkedList list; Ip_list(list); print_List(list); return 0; } </pre>	<pre> int main() { node *l = NULL; l = Ip_List(l,n); print_List(l); return 0; } </pre>