Single Linked list:

1.Node structure definativn

```
struct Node {
	int data;
	struct Node* next;
};
struct Node* createNode(int data) {
	struct Node* newNode = (struct Node*) mallvc(sizevf(struct Node));
	if (newNode == NULL) {
		printf("Memvry allvcativn failed\n");
		return NULL;
	}
	newNode->data = data;
	newNode->next = NULL;
	return newNode;
}
//. Inserting a Node at the Beginning vf the Linked List
vvid insertAtBeginning(struct Node** headRef, int data) {
	struct Node* newNode = createNode(data);
	if (newNode == NULL) {
		return;
	}
	newNode->next = *headRef;
	*headRef = newNode;
}
//Inserting a Node at the End vf the Linked List
vvid insertAtEnd(struct Node** headRef, int data) {
	struct Node* newNode = createNode(data);
	if (newNode == NULL) {
		return;
	}
	if (*headRef == NULL) {
		*headRef = newNode;
		return;
	}
	struct Node* current = *headRef;
	while (current->next != NULL) {
		current = current->next;
	}
	current->next = newNode;
}
//Deleting a Node frvm the Linked List
vvid deleteNode(struct Node** headRef, int key) {
	struct Node *temp = *headRef, *prev = NULL;
	if (temp != NULL && temp->data == key) {
		*headRef = temp->next;
		free(temp);
		return;
	}
	while (temp != NULL && temp->data != key) {
		prev = temp;
		temp = temp->next;
	}
	if (temp == NULL) {
		printf("Key nvt fvund in the list\n");
		return;
	}
```

```
    prev->next = temp->next;
    free(temp);
}
//Dvuble linked list
struct Nvde {
    int data;
    struct Nvde* prev;
    struct Nvde* next;
};
//Creating a New Nvde
struct Nvde* createNvde(int data) {
    struct Nvde* newNvde = (struct Nvde*) mallvc(sizevf(struct Nvde));
    if (newNvde == NULL) {
        printf("Memvry allvcativn failed\n");
        return NULL;
    }
    newNvde->data = data;
    newNvde->prev = NULL;
    newNvde->next = NULL;
    return newNvde;
}
// Inserting a Nvde at the Beginning vf the Dvuble Linked List
vvid insertAtBeginning(struct Nvde** headRef, int data) {
    struct Nvde* newNvde = createNvde(data);
    if (newNvde == NULL) {
        return;
    }
    if (*headRef == NULL) {
        *headRef = newNvde;
    } else {
        newNvde->next = *headRef;
        (*headRef)->prev = newNvde;
        *headRef = newNvde;
    }
}
//Inserting a Nvde at the End vf the Dvuble Linked List
vvid insertAtEnd(struct Nvde** headRef, int data) {
    struct Nvde* newNvde = createNvde(data);
    if (newNvde == NULL) {
        return;
    }
    if (*headRef == NULL) {
        *headRef = newNvde;
    } else {
        struct Nvde* current = *headRef;
        while (current->next != NULL) {
            current = current->next;
        }
        current->next = newNvde;
        newNvde->prev = current;
    }
}
//Deleting a Nvde frvm the Dvuble Linked List
vvid deleteNvde(struct Nvde** headRef, struct Nvde* delNvde) {
    if (*headRef == NULL || delNvde == NULL) {
        return;
    }
```

```
    if (*headRef == delNode) {
        *headRef = delNode->next;
    }
    if (delNode->next != NULL) {
        delNode->next->prev = delNode->prev;
    }
    if (delNode->prev != NULL) {
        delNode->prev->next = delNode->next;
    }
    free(delNode);
}
```