

## DLL

```
struct node {
```

```
    int data;
```

```
    struct node * next;
```

```
    struct node * prev;
```

```
};
```

```
struct node * next head;
```

```
void insertbeginning (int item) {
```

```
    struct node * ptr = (struct node *) malloc (sizeof (struct node));
```

```
    if (head == NULL) {
```

```
        ptr -> next = NULL;
```

```
        ptr -> prev = NULL;
```

```
        ptr -> data = item;
```

```
        head = ptr;
```

```
    } else {
```

```
        ptr -> data = item;
```

```
        ptr -> prev = NULL;
```

```
        ptr -> next = head;
```

```
        head -> prev = ptr;
```

```
        head = ptr;
```

```
    }
```

```
}
```

```
void deleteSpecific ( ) {
```

```
    struct node * ptr, * temp;
```

```
    int val;
```

```
    printf ("Enter the value: ");
```

```
    scanf ("%d", &val);
```

```
    temp = head;
```



```

while (temp → data != val) temp = temp → next;
if (temp → next == null) {
    printf ("In can't delete\n");
} else {
    ptr = temp → next;
    temp → next = ptr → next;
    ptr → next → prev = temp;
    free(ptr);
    printf ("Node Deleted\n");
}
}
}

```

```

void display() {
    struct node *ptr;
    ptr = head;
    if (ptr == NULL) {
        printf ("Empty\n");
    } else {
        while (ptr != NULL) {
            printf ("%d ", ptr → data);
            ptr = ptr → next;
        }
        printf ("\n");
    }
}
}
}

```



```
int main () {  
    int op = 0;  
    int a;  
    printf ("1. Insert at left 2. Delete at specified 3. Display\n");
```

```
while (op != 4) {
```

```
    printf ("Enter your choice: ");
```

```
    scanf ("%d", &op);
```

```
    switch (op) {
```

```
        case 1:
```

```
            printf ("Enter value to be inserted: ");
```

```
            scanf ("%d", &a);
```

```
            insertBeginning(a);
```

```
            break;
```

```
        case 2:
```

```
            deleteSpecified();
```

```
            break;
```

```
        case 3:
```

```
            display();
```

```
            break;
```

```
    }
```

```
    }
```

```
}
```