

Singly linked list

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  struct node
4  {
5      int data;
6      struct node *next;
7  };
8  struct node *head = NULL;
9  void append(int data)
10 {
11     struct node *newnode=malloc(sizeof(struct node));
12     newnode->data=data;
13     newnode->next=NULL;
14     if(head==NULL)
15     {
16         head=newnode;
17     }
18     else
19     {
20         struct node *temp=head;
21         while(temp->next!=NULL)
22         {
23             temp=temp->next;
24         }
25         temp->next = newnode;
26     }
27 }
28 void display()
```

```
28 void display()
29 {
30     struct node *temp=head;
31     while(temp!=NULL)
32     {
33         printf("%d ",temp->data);
34         temp=temp->next;
35     }
36 }
37 int main()
38 {
39     int n,data;
40     printf("enter number of node");
41     scanf("%d",&n);
42     for(int i=0;i<n;i++)
43     {
44         scanf("%d",&data);
45         if(data>0)
46         {
47             append(data);
48         }
49     }
50     display();
51 }
```

```
enter number of node5
10 20 30 40 50
10 20 30 40 50
```

Doubly linked list

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  struct node
4  {
5      int data;
6      struct node *next;
7      struct node *prev;
8  };
9  struct node *head=NULL;
10 void append(int data)
11 {
12     struct node *newnode=malloc(sizeof(struct node));
13     newnode->data=data;
14     newnode->prev=NULL;
15     newnode->next=NULL;
16     if(head==NULL)
17     {
18         head=newnode;
19     }
20     else
21     {
22         struct node *temp = head;
23         while(temp->next !=NULL)
24         {
25             temp=temp->next;
26         }
27         temp->next=newnode;
28         newnode->prev=temp;
29     }
```

```
26     }
27     temp->next=newnode;
28     newnode->prev=temp;
29 }
30 }
31 void display()
32 {
33     struct node *temp=head;
34     while(temp!=NULL)
35     {
36         printf("%d ",temp->data);
37         temp=temp->next;
38     }
39 }
40 int main()
41 {
42     int n,data;
43     printf("enter number of node:");
44     scanf("%d",&n);
45     for(int i=0;i<n;i++)
46     {
47         scanf("%d",&data);
48         if(data>0)
49         {
50             append(data);
51         }
52     }
53     display();
54 }
```

```
enter number of node:3
10 20 30
10 20 30
```

Printing in reverse

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  struct node
4  {
5      int data;
6      struct node *next; //creating structure
7      struct node *prev;
8  };
9  struct node *head=NULL;
10 void append(int data)
11 {
12     struct node *newNode=malloc(sizeof(struct node));
13     newNode->data=data;
14     newNode->next=NULL;
15     newNode->prev=NULL;
16     if(head==NULL)
17     {
18         head=newNode;
19     }
20     else
21     {
22
23         struct node *temp=head;
24         while(temp->next!=NULL)
25         {
26             temp=temp->next;
27         }
28         temp->next=newNode;
29         newNode->prev=temp;
```

```
29         newNode->prev=temp;
30     }
31 }
32 void display()
33 {
34     struct node *temp=head;
35     while(temp!=NULL)
36     {
37         printf("%d ",temp->data); //33
38         temp=temp->next;
39     }
40 }
41 void backward()
42 {
43     struct node *temp=head;
44     while(temp->next!=NULL)
45     {
46         temp=temp->next;
47     }
48     while(temp!=NULL)
49     {
50         printf(" %d",temp->data);
51         temp=temp->prev;
52     }
53 }
54 int main()
55 {
56     int n,data;
57     printf("Enter number of node:");
```

```

53 }
54 int main()
55 {
56     int n,data;
57     printf("Enter number of node:");
58     scanf("%d",&n);//5
59     for(int i=0;i<n;i++)
60     {
61         scanf("%d",&data);//10
62         if(data>0)
63         {
64             append(data);
65         }
66     }
67     printf("The data in the nodes are:");
68     display();
69     printf(" \n");
70     backward();
71 }

```

```

Enter number of node:3
10 20 30
The data in the nodes are:10 20 30
30 20 10

```

Circular linked list

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  struct node
4  {
5      int data;
6      struct node *next; //creating structure
7  };
8  struct node *head=NULL;
9  void append(int data)
10 {
11     struct node *newNode=malloc(sizeof(struct node));
12     newNode->data=data;
13     if(head==NULL)
14     {
15         head=newNode;
16         newNode->next=head;
17     }
18     else
19     {
20         struct node *temp=head;
21         while((temp->next!=head))
22         {
23             temp=temp->next;
24         }
25         temp->next=newNode;
26         newNode->next=head;
27     }
28 }
```

```
29 void display()
30 {
31     struct node *temp=head;
32     do
33     {
34         printf("%d ",temp->data);
35         temp=temp->next;
36     }while(temp!=head);
37 }
38 int main()
39 {
40     int n,data;
41     printf("Enter number of node:");
42     scanf("%d",&n);
43     for(int i=0;i<n;i++)
44     {
45         scanf("%d",&data);
46         if(data>0)
47         {
48             append(data);
49         }
50     }
51     printf("The data in the nodes are:");
52     display();
53 }
```

```
Enter number of node:3
10 20 30
The data in the nodes are:10 20 30
```

Stack implementation, reversing , inserting node at beginning

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  struct node
4  {
5      int data;
6      struct node *next;
7  };
8  struct node *head=NULL;
9  void insert(int data)
10 {
11     struct node*newnode=malloc(sizeof(struct node));
12     newnode -> data=data;
13     newnode ->next=head;
14     head=newnode;
15 }
16 void display()
17 {
18     struct node*temp=head;
19     while(temp!=NULL)
20     {
21         printf("%d |",temp->data);
22         temp=temp->next;
23     }
24 }
```

```
25 int main()
26 {
27     int n,data;
28     printf("enter number of node");
29     scanf("%d",&n);
30     for(int i=0;i<n;i++)
31     {
32         scanf("%d",&data);
33         if(data>0)
34         {
35             insert(data);
36         }
37     }
38     display();
39     return 0;
40 }
```

enter number of node3
10 20 30
30 20 10

Inserting element at second position

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  struct node
4  {
5      int data;
6      struct node *next;
7  };
8  struct node *head = NULL;
9  void append(int data)
10 {
11     struct node *newnode = malloc(sizeof(struct node));
12     newnode->data = data;
13     newnode->next = NULL;
14     if (head == NULL)
15     {
16         head = newnode;
17     }
18     else
19     {
20         struct node *temp = head;
21         while (temp->next != NULL)
22         {
23             temp = temp->next;
24         }
25         temp->next = newnode;
26     }
27 }
28 void insert_second(int data)
29 {
30     struct node *newnode = malloc(sizeof(struct node));
31     if (head == NULL)
32     {
33         printf("List is empty, cannot insert at second position\n");
34         return;
35     }
36     newnode->data = data;
37     newnode->next = head->next;
38     head->next = newnode;
39 }
40 void display()
41 {
42     struct node *temp = head;
43     while (temp != NULL)
44     {
45         printf("%d ", temp->data);
46         temp = temp->next;
47     }
48 }
```

```

49 int main()
50 {
51     int n, data;
52     printf("enter number of nodes: ");
53     scanf("%d", &n);
54     for (int i = 0; i < n; i++)
55     {
56         scanf("%d", &data);
57         append(data);
58     }
59     printf("Enter data to insert at second position: ");
60     scanf("%d", &data);
61     insert_second(data);
62     printf("Linked List: ");
63     display();
64     return 0;
65 }
66

```

input

```

enter number of nodes: 3
10 30 40
Enter data to insert at second position: 20
Linked List: 10 20 30 40

```