

## 1.)Basic linked list

```
main.c | Run | Output | Share | Run | Output  
1 #include <stdio.h>  
2 #include <stdlib.h>  
3 struct Node {  
4     int data;  
5     struct Node* next;  
6 };  
7  
8 int main() {  
9     struct Node *head, *second, *third;  
10  
11     head = (struct Node*)malloc(sizeof(struct Node));  
12     second = (struct Node*)malloc(sizeof(struct Node));  
13     third = (struct Node*)malloc(sizeof(struct Node));  
14  
15     head->data = 15;  
16     head->next = second;  
17  
18     second->data = 20;  
19     second->next = third;  
20  
21     third->data = 35;  
22     third->next = NULL;  
23  
24     struct Node* temp = head;  
25     while (temp != NULL) {  
15 20 35  
==== Code Execution Successful ===|
```

```
#include <stdio.h>  
  
#include <stdlib.h>  
  
struct Node {  
  
    int data;  
  
    struct Node* next;  
};  
  
  
int main() {  
  
    struct Node *head, *second, *third;  
  
  
    head = (struct Node*)malloc(sizeof(struct Node));  
    second = (struct Node*)malloc(sizeof(struct Node));  
    third = (struct Node*)malloc(sizeof(struct Node));  
  
  
    head->data = 15;  
    head->next = second;  
  
  
    second->data = 20;  
    second->next = third;  
  
  
    third->data = 35;  
    third->next = NULL;
```

```
struct Node* temp = head;  
while (temp != NULL) {  
    printf("%d ", temp->data);  
    temp = temp->next;  
}  
  
printf("\n");  
return 0;  
}
```

**OUTPUT:**

15 20 35

## 2.) Insert and start

The screenshot shows a code editor interface with a dark theme. The left pane contains the source code in 'main.c', and the right pane shows the 'Output' of the executed code.

```
main.c
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 struct Node {
5     int data;
6     struct Node* next;
7 };
8
9 struct Node* insertAtStart(struct Node* head, int value) {
10    struct Node* new_node = (struct Node*)malloc(sizeof(struct Node));
11    new_node->data = value;
12    new_node->next = head;
13    return new_node;
14 }
15
16 int main() {
17    struct Node *head, *second, *third;
18
19    head = (struct Node*)malloc(sizeof(struct Node));
20    second = (struct Node*)malloc(sizeof(struct Node));
21    third = (struct Node*)malloc(sizeof(struct Node));
22
23    head->data = 15;
24    head->next = second;
25 }
```

Output:

```
746 15 20 35
== Code Execution Successful ==
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
struct Node {
```

```
    int data;
```

```
    struct Node* next;
```

```
};
```

```
struct Node* insertAtStart(struct Node* head, int value) {
```

```
    struct Node* new_node = (struct Node*)malloc(sizeof(struct Node));
```

```
    new_node->data = value;
```

```
    new_node->next = head;
```

```
    return new_node;
```

```
}
```

```
int main() {
```

```
    struct Node *head, *second, *third;
```

```
    head = (struct Node*)malloc(sizeof(struct Node));
```

```
    second = (struct Node*)malloc(sizeof(struct Node));
```

```
    third = (struct Node*)malloc(sizeof(struct Node));
```

```
head->data = 15;  
head->next = second;  
  
second->data = 20;  
second->next = third;  
  
third->data = 35;  
third->next = NULL;  
  
head = insertAtStart(head, 746);  
  
struct Node* temp = head;  
while (temp != NULL) {  
    printf("%d ", temp->data);  
    temp = temp->next;  
}  
  
printf("\n");  
return 0;  
}
```

## **OUTPUT:**

746 15 20 35

### 3.) Insert and end

The screenshot shows a code editor interface with a dark theme. The left pane displays the source code in a file named 'main.c'. The right pane shows the output of the code execution.

```
main.c | [ ] Share Run | Output  
1 #include <stdio.h>  
2 #include <stdlib.h>  
3  
4 struct Node {  
5     int data;  
6     struct Node* next;  
7 };  
8  
9 struct Node* insertAtEnd(struct Node* head, int value) {  
10     struct Node* new_node = (struct Node*)malloc(sizeof(struct Node));  
11     new_node->data = value;  
12     new_node->next = NULL;  
13  
14     if (head == NULL)  
15         return new_node;  
16  
17     struct Node* temp = head;  
18     while (temp->next != NULL) {  
19         temp = temp->next;  
20     }  
21  
22     temp->next = new_node;  
23     return head;  
24 }
```

Output:  
15 20 35 622  
== Code Execution Successful ==

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
struct Node {
```

```
    int data;
```

```
    struct Node* next;
```

```
};
```

```
struct Node* insertAtEnd(struct Node* head, int value) {
```

```
    struct Node* new_node = (struct Node*)malloc(sizeof(struct Node));
```

```
    new_node->data = value;
```

```
    new_node->next = NULL;
```

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

```
        return new_node;
```

```
    struct Node* temp = head;
```

```
    while (temp->next != NULL) {
```

```
        temp = temp->next;
```

```
}
```

```
temp->next = new_node;
return head;
}

int main() {
    struct Node *head, *second, *third;

    head = (struct Node*)malloc(sizeof(struct Node));
    second = (struct Node*)malloc(sizeof(struct Node));
    third = (struct Node*)malloc(sizeof(struct Node));

    head->data = 15;
    head->next = second;

    second->data = 20;
    second->next = third;

    third->data = 35;
    third->next = NULL;

    head = insertAtEnd(head, 622);

    struct Node* temp = head;
    while (temp != NULL) {
        printf("%d ", temp->data);
        temp = temp->next;
    }

    printf("\n");
    return 0;
}
```

**OUTPUT:**

15 20 35 622

#### 4.)Delete from start

The screenshot shows a code editor interface with a dark theme. The left pane contains the source code for `main.c`. The right pane shows the output of the code execution.

```
main.c
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 struct Node {
5     int data;
6     struct Node* next;
7 };
8
9 void deleteFromStart(struct Node** head) {
10    if (*head == NULL)
11        return;
12
13    struct Node* temp = *head;
14    *head = (*head)->next;
15    free(temp);
16 }
17
18 int main() {
19     struct Node *head, *second, *third;
20
21     head = (struct Node*)malloc(sizeof(struct Node));
22     second = (struct Node*)malloc(sizeof(struct Node));
23     third = (struct Node*)malloc(sizeof(struct Node));
24 }
```

Output:

```
20 35
== Code Execution Successful ==
```

```
#include <stdio.h>

#include <stdlib.h>

struct Node {

    int data;

    struct Node* next;

};

void deleteFromStart(struct Node** head) {

    if (*head == NULL)

        return;

    struct Node* temp = *head;

    *head = (*head)->next;

    free(temp);

}

int main() {

    struct Node *head, *second, *third;

    head = (struct Node*)malloc(sizeof(struct Node));
```

```
second = (struct Node*)malloc(sizeof(struct Node));  
third = (struct Node*)malloc(sizeof(struct Node));  
  
head->data = 15;  
head->next = second;  
  
second->data = 20;  
second->next = third;  
  
third->data = 35;  
third->next = NULL;  
  
deleteFromStart(&head);  
  
struct Node* temp = head;  
while (temp != NULL) {  
    printf("%d ", temp->data);  
    temp = temp->next;  
}  
  
printf("\n");  
return 0;  
}
```

## **OUTPUT:**

20 35