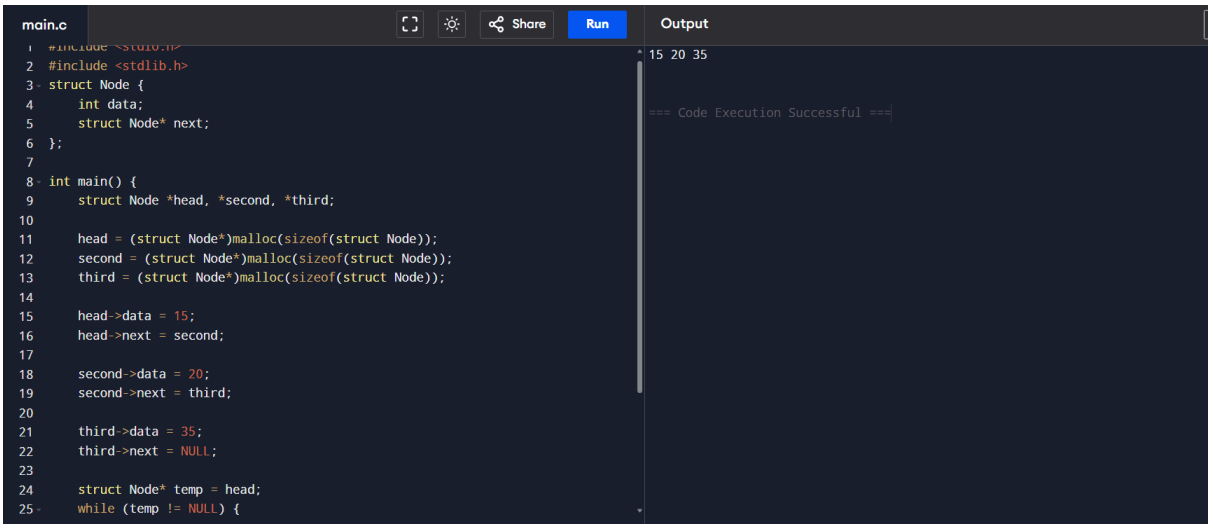
1.)Basic linked list



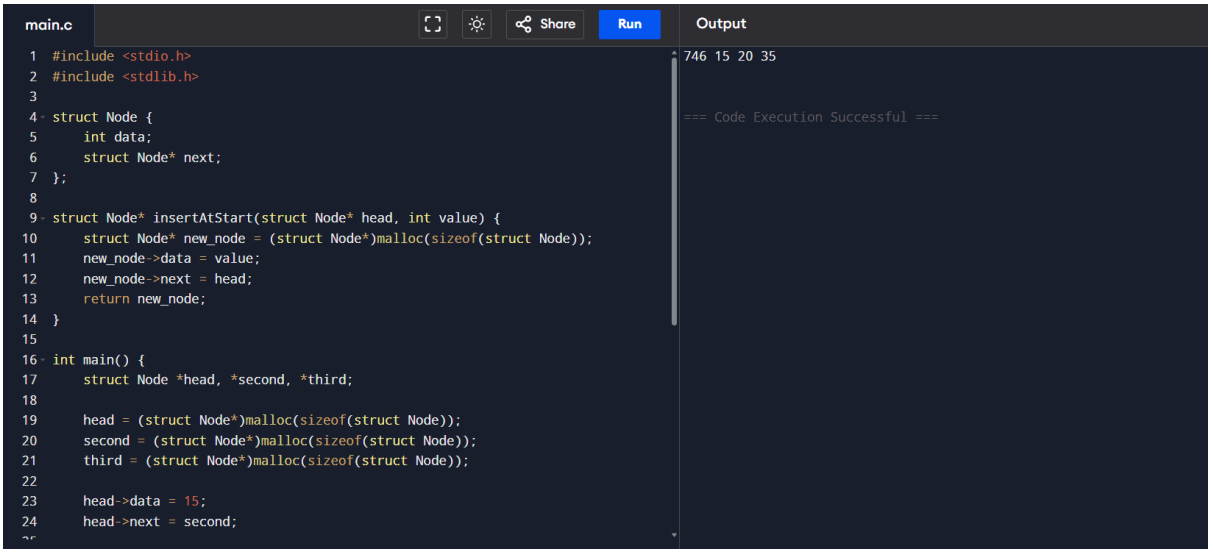

```
#include <stdio.h>
#include <stdlib.h>
struct Node {
    int data;
    struct Node* next;
};

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) {
```

```
    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



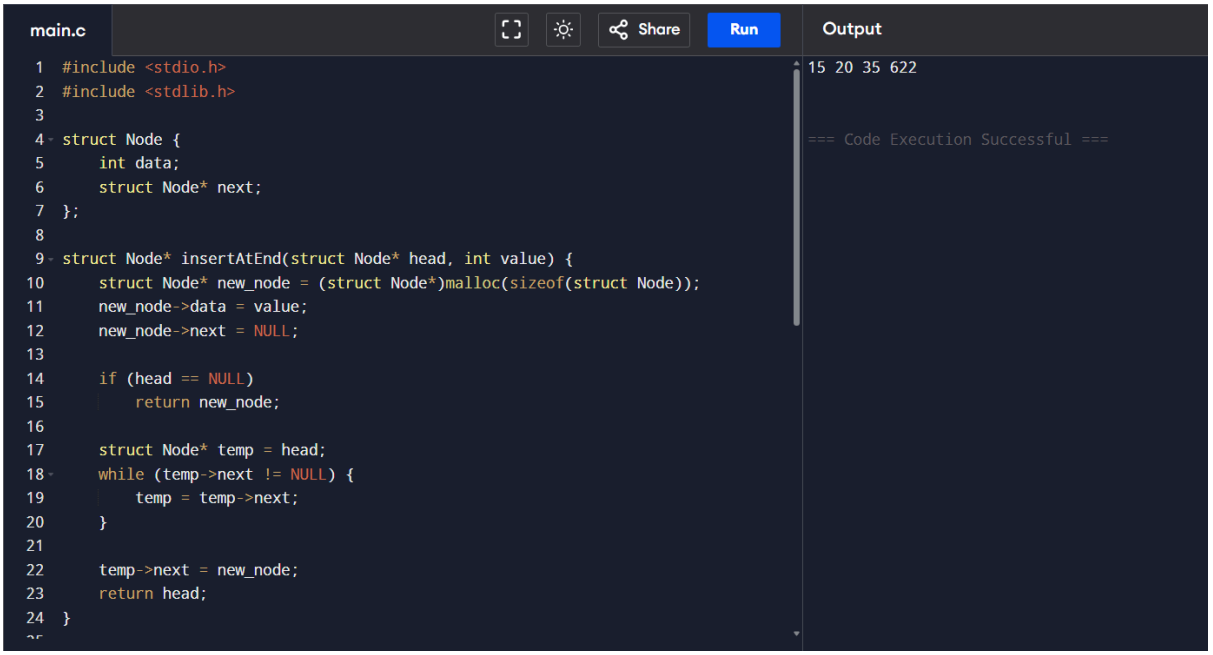

```
#include <stdio.h>
#include <stdlib.h>

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



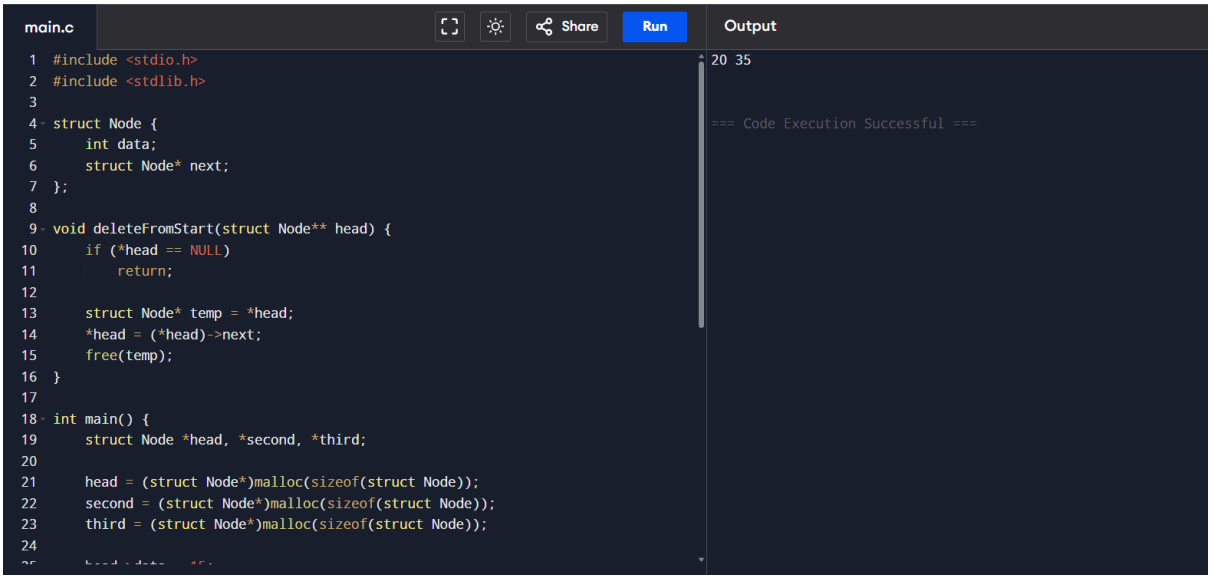

```
#include <stdio.h>

#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




```
#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));
```

```
    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