Name: Mithlesh Yeole

Roll no. : B3-B3-59

Practical 4A

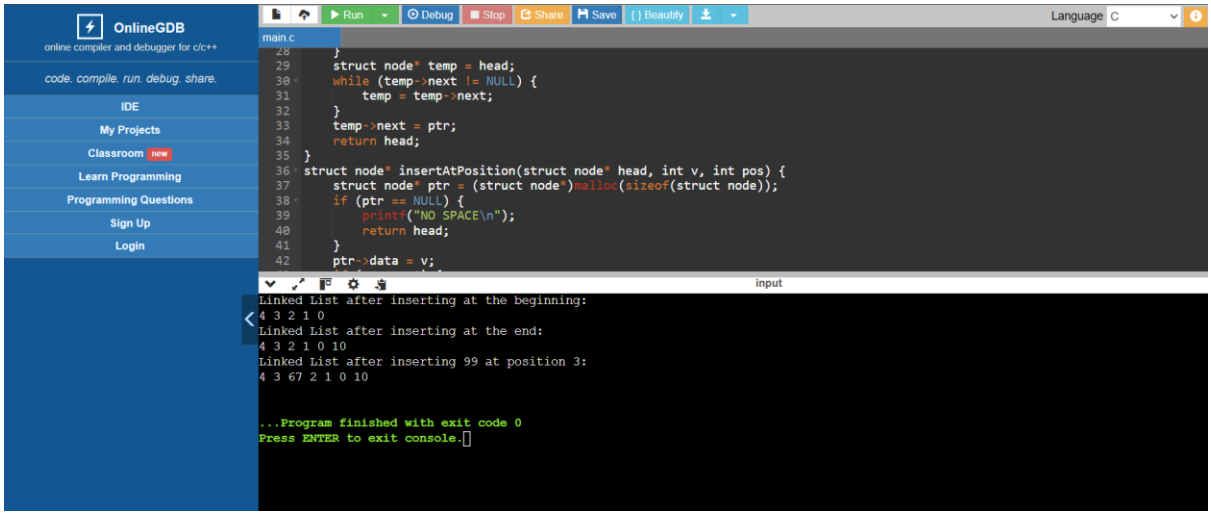
Code:

```
#include <stdio.h>
#include <stdlib.h>
struct node {
   int data;
   struct node* next;
};
struct node* insertbegin(struct node* head, int v) {
   struct node* ptr = (struct node*)malloc(sizeof(struct node));
   if (ptr == NULL) {
      printf("NO SPACE\n");
      return head;
   }
   ptr->data = v;
   ptr->next = head;
   head = ptr;
   return head;
}
struct node* insertend(struct node* head, int v) {
   struct node* ptr = (struct node*)malloc(sizeof(struct node));
   if (ptr == NULL) {
      printf("NO SPACE\n");
      return head;
   }
   ptr->data = v;
   ptr->next = NULL;
   if (head == NULL) {
      return ptr;
```

```
    }
    struct node* temp = head;
    while (temp->next != NULL) {
        temp = temp->next;
    }
    temp->next = ptr;
    return head;
}
struct node* insertAtPosition(struct node* head, int v, int pos) {
    struct node* ptr = (struct node*)malloc(sizeof(struct node));
    if (ptr == NULL) {
        printf("NO SPACE\n");
        return head;
    }
    ptr->data = v;
    if (pos == 1) {
        ptr->next = head;
        return ptr;
    }
    struct node* temp = head;
    for (int i = 1; i < pos - 1 && temp != NULL; i++) {
        temp = temp->next;
    }
    if (temp == NULL) {
        printf("Invalid position!\n");
        free(ptr);
        return head;
    }
    ptr->next = temp->next;
    temp->next = ptr;
    return head;
}
```

```
void traverse(struct node* head) {
    struct node* ptr = head;
    while (ptr != NULL) {
        printf("%d ", ptr->data);
        ptr = ptr->next;
    }
    printf("\n");
}
int main() {
    struct node* head = NULL;
    for (int i = 0; i < 5; i++) {
        head = insertbegin(head, i);
    }
    printf("Linked List after inserting at the beginning:\n");
    traverse(head);
    head = insertend(head, 10);
    printf("Linked List after inserting at the end:\n");
    traverse(head);
    head = insertAtPosition(head, 67, 3);
    printf("Linked List after inserting 99 at position 3:\n");
    traverse(head);
    return 0;
}
```

Output:


OnlineGDB
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
Run
Debug
Stop
Share
Save
{ } Beautify
Language C
main.c
}
struct node* temp = head;
while (temp->next != NULL) {
temp = temp->next;
}
temp->next = ptr;
return head;
}
struct node* insertAtPosition(struct node* head, int v, int pos) {
struct node* ptr = (struct node*)malloc(sizeof(struct node));
if (ptr == NULL) {
printf("NO SPACE\n");
return head;
}
ptr->data = v;
input
Linked List after inserting at the beginning:
4 3 2 1 0
Linked List after inserting at the end:
4 3 2 1 0 10
Linked List after inserting 99 at position 3:
4 3 67 2 1 0 10
...Program finished with exit code 0
Press ENTER to exit console.