Run | Debug | Stop | Share | Save | { } Beautify | Language C

main.c

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

struct Node{
    int data;
    struct Node *next;
};


void linkedListTraversal(struct Node* ptr)
{
    while(ptr!=NULL)
    {
        printf("%d\t", ptr->data);
        ptr = ptr->next;
    }
}

struct Node* insertionAtFront(struct Node* head,int data){
    struct Node * ptr=(struct Node*)malloc(sizeof(struct Node));
    ptr->data=data;
    ptr->next=head;
    head=ptr;
    return head;
}

struct Node* insertionAtEnd(struct Node* head,int data){
    struct Node* ptr=(struct Node*)malloc(sizeof(struct Node));
    ptr->data=data;
    struct Node* new=head;

    while(new->next!=NULL)
    {
        new=new->next;
    }
    new->next=ptr;
    ptr->next=NULL;
    return head;
}
struct Node* insertionAtIndex(struct Node* head,int data,int index)
{
    struct Node* ptr=(struct Node*)malloc(sizeof(struct Node));
    struct Node*new2=head;
    int i=0;

    while(i!=index-1)
    {
```

```
    {
        new2=new2->next;
        i++;
    }
    ptr->data=data;
    ptr->next=new2->next;
    new2->next=ptr;
    return head;
}

void main()
{
    struct Node* head;
    struct Node* second;
    struct Node* third;
    struct Node* fourth;
    struct Node* fifth;
    head=(struct Node*)malloc(sizeof(struct Node));
    second=(struct Node*)malloc(sizeof(struct Node));
    third=(struct Node*)malloc(sizeof(struct Node));
    fourth=(struct Node*)malloc(sizeof(struct Node));
    fifth=(struct Node*)malloc(sizeof(struct Node));

    head->data=100;
    head->next=second;
    second->data=99;
    second->next=third;
    third->data=98;
    third->next=fourth;
    fourth->data=97;
    fourth->next=fifth;
    fifth->data=96;
    fifth->next=NULL;

    printf("Original linked list:\n");
    linkedListTraversal(head);
    printf("\nLinked list after inserting element in front\n");
    head=insertionAtFront(head,21);
    linkedListTraversal(head);
    printf("\nLinked list after inserting element in end\n");
    head=insertionAtEnd(head,22);
    linkedListTraversal(head);
    printf("\nLinked list after inserting element at 3rd position\n");
    head=insertionAtIndex(head,23,3);
    linkedListTraversal(head);
}
```

input

```
Original linked list:
100     99      98      97      96
Linked list after inserting element in front
21      100     99      98      97      96
Linked list after inserting element in end
21      100     99      98      97      96      22
Linked list after inserting element at 3rd position
21      100     99      23      98      97      96      22

...Program finished with exit code 0
Press ENTER to exit console.
```