11/1/2024

Singly linked list

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

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

void push( struct Node** head_ref, int new_data);
void append( struct Node** head_ref, int new_data);
void insert_at_first(struct Node **list_ref, int new_data);

struct Node* createNode(int value) {
    struct Node* newnode=(struct Node*)malloc(
        sizeof(struct Node));
    newnode->data = data;
    newnode->next =NULL;
    return newnode;
}
void insertatbeginning (struct Node** head,int data){
    struct Node* newnode = createnode(data);
    new Node->next = *head;
    *head = newnode;
}
```

```
void insertatBeginning (struct Node **head, int data)
{
    struct Node* newnode = createNode(data);
    newnode -> next = *head;
    *head = newNode;
}
void insertatposition (struct Node **head, int data)
    if (position <= 0) {
        printf ("Invalid position, please enter a
        position greater than 0\n");
        return;
    }
struct Node* newnode = createNode (data);
if (position == 1) {
        newNode -> next = *head;
        *head = newNode;
        return;
    }
int main() {
        struct Node *head = NULL;
        int n, element;

        printf ("Enter no of elements:");
        scanf ("%d", &n);
```

```
for(int i=0; i<n; i++) {
    printf("Enter element %d:", i+1);
    scanf("%d", &element);
    insertAtend(&head, element);
}

int newData;
printf("Enter a new element at beginning:");
scanf("%d", &newData);
printf("Linked list after inserting at beginning:");
display(head);

int position;
printf("Enter the position to new element:");
scanf("%d", &position);
printf("insert into position:");
scanf("%d", &newData);
printf("Enter a new element to insert at the el");
scanf("%d", &newData);
display(head);
return 0;
}
```

Output:- Enter the number of elements :3

element 1 : 24

element 2 : 76

element 3 : 34

Linked list : 24 → 76 → 74 → NULL
Enter a new element to insert at the beginning : 12
Enter the position to insert a new element : 12
Enter the new after insertion at end : 12 → 65 → 24 → 76 → 74 → NULL