```c
// Implementation of singly linked list
#include <stdio.h>
#include <stdlib.h>

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

struct Node* createNode(int data) {
    struct Node* newNode = (struct Node*) malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = NULL;
    return newNode;
}

void insertAtFirst(struct Node** head, int data) {
    struct Node* newNode = createNode(data);
    newNode->next = *head;
    *head = newNode;
}

void insertAtEnd(struct Node** head, int data) {
    struct Node* newNode = createNode(data);
    if (*head == NULL) {
        *head = newNode;
    } else {
        struct Node* temp = *head;
        while (temp->next != NULL) {
            temp = temp->next;
        }
        temp->next = newNode;
    }
}

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

int main() {
```

```c
    struct Node* head = NULL;
    int choice, data;
    while (1) {
        printf("Enter 1 to insert at the beginning, 2 to insert at the end, 3 to
display, 4 to exit: ");
        scanf("%d", &choice);
        switch (choice) {
            case 1:
                printf("Enter data: ");
                scanf("%d", &data);
                insertAtFirst(&head, data);
                break;
            case 2:
                printf("Enter data: ");
                scanf("%d", &data);
                insertAtEnd(&head, data);
                break;
            case 3:
                display(head);
                break;
            case 4:
                exit(0);
            default:
                printf("Invalid choice\n");
        }
    }
    return 0;
```

Output:

```
soft-MIEngine-in-jx35qv1b.upz --stdout=Microsoft-MIEngine-Out-ofm531rc.ri4 --stderr=Microsof
3hd5yswm.552 --dbgExe=C:\\msys64\\ucrt64\\bin\\gdb.exe --interpreter=mi
Enter 1 to insert at the beginning, 2 to insert at the end, 3 to display, 4 to exit: 1
Enter data: 2
Enter 1 to insert at the beginning, 2 to insert at the end, 3 to display, 4 to exit: 3
2
Enter 1 to insert at the beginning, 2 to insert at the end, 3 to display, 4 to exit: 1
Enter data: 5
Enter 1 to insert at the beginning, 2 to insert at the end, 3 to display, 4 to exit: 3
5 2
Enter 1 to insert at the beginning, 2 to insert at the end, 3 to display, 4 to exit: 2
Enter data: 6
Enter 1 to insert at the beginning, 2 to insert at the end, 3 to display, 4 to exit: 3
5 2 6
Enter 1 to insert at the beginning, 2 to insert at the end, 3 to display, 4 to exit:
```

PAGE :
DATE : / /

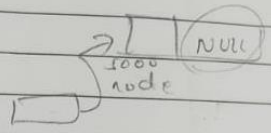
Implement singly linked list

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

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

struct Node* creatNode ( int data)
    {
        struct Node* newnode = (struct Node*) malloc (sizeof (struct Node));
        newnode -> data = data;
        newnode -> next = NULL;
    }   return newNode;

void insertatfirst ( struct Node **head, int data)
    {
        struct Node* newNode = (creatNode ( data );
        newnode -> next = *head;
        *head = newNode
    }
void insert at end ( struct Node** head, int data)
    {
        struct Node* newNode = creatNode (data);
        if (*head == NULL)
        {
            *head = newnode;
        }
        else
        {
        struct Node* temp = *head;
```

PAGE :
DATE : / /

~~for(int i=0; i<~~

```
while (temp -> next != NULL)
 {
    temp = temp -> next;
 }
 temp -> next = newnode;
 new->node->next = NULL;
}
```




```
void display (Struct Node ** head)
  {
    Struct Node * temp = *head;
    while (temp != NULL)
     {
       printf("%d ", temp -> data);
     temp = temp -> next;
     }
     printf ("%d", temp -> data);
  }
void main(*)
{  Struct Node* head = NULL
   printf("Enter int d;
   while (1)
    { printf(" Enter 1 to insertatfirst, 2 to insertatend,
              3 to exit")
      int choice = 0;
      scanf("%d", &choice);
      if (choice == 1)
       {
          printf("Enter data");
          int d;
          scanf("%d", &d);
          insertatfirst (&head, d);
       }
```

```
else if (choice == 2)
{
    printf("Enter data:");
    scanf("%d", &d)
    insertatend (&head, d);
}
else if (choice == 3)
{
    display(&head);
}
else
    break;
}
```

Output:

1. Insert at first | 2. Insert at End | 3. Display | 4. Exit
1
Enter data : 2
1. Insert at first | 2. . .
3
2
1. Insert at first | 2. .
1
Enter data : 5
1. Insert at first | 2. -
3
5,2
Insert at first | 2. -
2
Enter data: 6
5 2 6