

linked list:

22/01/2024

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

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

struct node *start = NULL;
struct node *create_ll(struct node *);
struct node *display(struct node *);
struct node *insert_beg(struct node *);
struct node *insert_end(struct node *);
struct node *insert_before(struct node *);
struct node *insert_after(struct node *);
struct node *insert_delete_beg(struct node *);
struct node *delete_end(struct node *);
struct node *delete_node(struct node *);
struct node *delete_after(struct node *);
struct node *delete_list(struct node *);
struct node *sort_list(struct node *);

int main()
{
    int choice;
    printf("1. create a list", "2. display", "3. insert_beg",
        "4. insert_end", "5. insert_before", "6. insert_after");
    printf("7. del_beg", "8. del_end", "9. del node", "10. del_after",
        "11. del_list", "12. sort_list", "13. exit");
    scanf("%d", &choice);
    switch(choice)
    {
        case 1: start = create_ll(start);
                printf("linked list created");
                break;
        case 3: start = insert_beg(start);
                break;
    }
```



```
Case 4: start = insert_end(start);  
break;
```

```
}  
} while(choice != 13);
```

```
return 0;
```

```
}  
struct node * create_ll(struct node *start)
```

```
{  
    struct node *new_node, *ptr;  
    int num;
```

```
    printf("enter -1 to end");
```

```
    printf("enter the data:");
```

```
    scanf("%d", &num);
```

```
    while(num != -1)
```

```
{  
    new_node = (struct node *) malloc (sizeof (struct node));
```

```
    new_node->data = num;
```

```
    if (start == NULL)
```

```
{  
        new_node->next = NULL
```

```
        start = new_node;
```

```
    }
```

```
    else
```

```
{  
        ptr = start;
```

```
        while (ptr->next != NULL)
```

```
            ptr = ptr->next;
```

```
            ptr->next = new_node;
```

```
            new_node->next = NULL;
```

```
        }
```

```
        printf("enter the data");
```

```
        scanf("%d", &num);
```

```
    }  
    return start;
```

```
};
```



```
struct node * display(struct node * start)
```

```
{ struct node * ptr;
```

```
ptr = start;
```

```
while(ptr != NULL)
```

```
{ printf("\t %d", ptr->data);
```

```
ptr = ptr->next;
```

```
}
```

```
return start;
```

```
};
```

```
struct node * insert_beg(struct node * start)
```

```
{ struct node * new-node;
```

```
int num;
```

```
printf("enter data: ");
```

```
scanf("%d", &num);
```

```
new-node = (struct node *) malloc(sizeof(struct
```

```
node));
```

```
new-node->data = num;
```

```
new-node->next = start;
```

```
start = new-node;
```

```
return start;
```

```
};
```

```
struct node * insert_end(struct node * start)
```

```
{ struct node * new-node, * ptr;
```

```
int num;
```

```
printf("enter data: ");
```

```
scanf("%d", &num);
```

```
new-node = (struct node *) malloc(sizeof(struct
```

```
node));
```

```
new-node->data = num;
```

```
new-node->next = NULL;
```

```
ptr = start;
```

```
if(start == NULL)
```

```
{ start = new-node;
```

```
}
```

```
else {
```



```

while(ptr->next != NULL)
    ptr = ptr->next;
ptr->next = new_node;
return start;
}

};

struct node *delete_beg(struct node *start)
{
    struct node *ptr;
    ptr = start;
    start = start->next;
    free(ptr);
    return start;
}

struct node *delete_end(struct node *start)
{
    struct node *ptr, *preptr;
    ptr = start;
    while(ptr->next != NULL)
    {
        preptr = ptr;
        ptr = ptr->next;
    }
    preptr->next = NULL;
    free(ptr);
    return start;
}

struct node *delete_list(struct node *start)
{
    struct node *ptr;
    if(start != NULL)
    {
        ptr = start;
        while(ptr != NULL)
        {
            printf("%d is deleted", ptr->data);
            ptr = start;
        }
    }
    return start;
}
};

```



```
struct node *sort_list(struct node *start)
```

```
{ struct node *ptr1, *ptr2;
```

```
int temp;
```

```
ptr1 = start;
```

```
while(ptr1->next != NULL)
```

```
{ ptr2 = ptr1->next;
```

```
while(ptr2 != NULL)
```

```
{ if(ptr1->data > ptr2->data)
```

```
{ temp = ptr1->data;
```

```
ptr1->data = ptr2->data;
```

```
ptr2->data = temp;
```

```
}
```

```
ptr2 = ptr2->next;
```

```
}
```

```
ptr1 = ptr1->next;
```

```
}
```

```
display(start);
```

```
};
```

P.E. *Expt 1*
22/1/24