

SEARCH UNSORTED

```
    /****      Program to Search in an Unsorted Linked List      ****/  
  
#include <stdio.h>  
  
void insert_last();  
void search_unsorted();  
void display();  
  
struct node  
{  
    int info;  
    struct node *link;  
} *start=NULL;  
  
int item;  
  
main()  
{  
    int ch;  
  
    do  
    {  
        printf("\n\n\n1. Insert Last\n2. Search Unsorted\n3. Display\n4. Exit\n");  
        printf("\nEnter your choice: ");  
        scanf("%d", &ch);  
  
        switch(ch)  
        {  
            case 1:  
                insert_last();  
                break;  
  
            case 2:  
                search_unsorted();  
                break;  
  
            case 3:  
                display();  
                break;  
  
            case 4:  
                exit(0);  
        }  
    }  
}
```

```
        default:
            printf("\n\nInvalid choice: Please try again.\n");
        }
    } while (1);
}

void insert_last()
{
    struct node *ptr;

    printf("\n\nEnter item: ");
    scanf("%d", &item);

    if(start == NULL)
    {
        start = (struct node *)malloc(sizeof(struct node));
        start->info = item;
        start->link = NULL;
    }
    else
    {
        ptr = start;

        while (ptr->link != NULL)
            ptr = ptr->link;

        ptr->link = (struct node *)malloc(sizeof(struct node));
        ptr = ptr->link;
        ptr->info = item;
        ptr->link = NULL;
    }

    printf("\nItem inserted: %d\n", item);
}

void search_unsorted()
{
    struct node *ptr = start;
    int loc = 1;

    printf("\n\nEnter ITEM to be searched: ");
    scanf("%d", &item);

    while (ptr != NULL)
    {
```

```
    if (item == ptr->info)
    {
        printf("\n\nItem %d is present at location %d\n", item,
            loc);
        return;
    }
    else
    {
        ptr = ptr->link;
        loc++;
    }
}

printf("\n\nItem is not present in the list\n");
}

void display()
{
    struct node *ptr = start;
    int i=1;

    if (ptr == NULL)
        printf("\nLinklist is empty.\n");
    else
    {
        printf("\nSr. No.\t\tAddress\t\tInfo\t\tLink\n");

        while(ptr != NULL)
        {
            printf("\n%d.\t\t%d\t\t%d\t\t%d\n", i, ptr, ptr->info,
                ptr->link);

            ptr = ptr->link;
            i++;
        }
    }
}
```