INSERT SORTED

```
     /****    Program to Insert Node in a Sorted Linked List    ****/

#include <stdio.h>

void insert_sorted();
void display();

struct node
{
     int info;
     struct node *link;
} *start=NULL;

int item;

main()
{
      int ch;
      do
      {
          printf("\n\n\n1. Insert Sorted\n2. Display\n3. Exit\n");
          printf("\nEnter your choice: ");
          scanf("%d", &ch);

          switch(ch)
          {
               case 1:
                    insert_sorted();
                    break;

               case 2:
                    display();
                    break;

               case 3:
                    exit(0);

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

```
void insert_sorted()
{
     struct node *ptr, *prev;

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

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

          while (ptr != NULL)
          {
              if (item < ptr->info)
              {
                   prev->link = (struct node *)malloc(sizeof(struct
                                                              node));
                   prev = prev->link;
                   prev->info = item;
                   prev->link = ptr;
                   return;
              }
              else if (ptr->link == NULL)
              {
                   ptr->link = (struct node *)malloc(sizeof(struct
                                                              node));
                   ptr = ptr->link;
                   ptr->info = item;
                   ptr->link = NULL;
                   return;
              }
```

```
                else
                {
                     prev = ptr;
                     ptr = ptr->link;
                }
           }
     }
}

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++;
          }
     }
}
```