REVERSE

```
            /****    Program to Reverse a Linked List    ****/

#include <stdio.h>

void insert_last();
void reverse();
void display();

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

int item;

main()
{
      int ch;

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

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

               case 2:
                    reverse();
                    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 reverse()
{
     struct node *ptr = start, *prev = NULL, *rev;

     while (ptr != NULL)
     {
           rev = prev;
           prev = ptr;
           ptr = ptr->link;
           prev->link = rev;
     }
```

```
    start = prev;
    printf("\n\nReversed Linked is:\n\n");
    display();
}

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++;
        }
    }
}
```