Doubly Linked List Program

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
   struct node *prev;
   struct node *next;
   int data;
};
struct node *start;
void insertion_beginning();
void insertion_last();
void deletion_beginning();
void deletion_last();
void display();
void main ()
{
int choice =0;
   while(choice != 9)
   {
      printf("\n*********Main Menu*********\n");
      printf("\nChoose one option from the following list ...\n");
      printf("\n===============================================\n");
      printf("\n1.Insert in begining\n2.Insert at last\n3.Delete from Beginning\n   4.Delete from last\n5.Show\n6.Exit\n");
      printf("\nEnter your choice?\n");
      scanf("\n%d",&choice);
      switch(choice)
      {
            case 1:
            insertion_beginning();
            break;
            case 2:
                   insertion_last();
            break;
            case 3:
            deletion_beginning();
            break;
            case 4:
            deletion_last();
            break;
             case 5:
            display();
            break;
            case 6:
            exit(0);
            break;
            default:
            printf("Please enter valid choice..");
        }
    }
}
void insertion_beginning()
{
   struct node *ptr;
   int item;
   ptr = (struct node *)malloc(sizeof(struct node));
   if(ptr == NULL)
   {
       printf("\nOVERFLOW");
```

```
  }
  else
  {
   printf("\nEnter Item value");
   scanf("%d",&item);

  if(start==NULL)
  {
      ptr->next = NULL;
      ptr->prev=NULL;
      ptr->data=item;
      start=ptr;
  }
  else
  {
      ptr->data=item;
      ptr->prev=NULL;
      ptr->next = start;
      start->prev=ptr;
      start=ptr;
  }
  printf("\nNode inserted\n");
}

}
void insertion_last()
{
  struct node *ptr,*temp;
  int item;
  ptr = (struct node *) malloc(sizeof(struct node));
  if(ptr == NULL)
  {
      printf("\nOVERFLOW");
  }
  else
  {
      printf("\nEnter value");
      scanf("%d",&item);
       ptr->data=item;
      if(start == NULL)
      {
          ptr->next = NULL;
          ptr->prev = NULL;
          start = ptr;
      }
      else
      {
         temp = start;
         while(temp->next!=NULL)
         {
             temp = temp->next;
         }
         temp->next = ptr;
         ptr ->prev=temp;
         ptr->next = NULL;
         }

      }
    printf("\nnode inserted\n");
   }
void deletion_beginning()
{
   struct node *ptr;
   if(head == NULL)
   {
       printf("\n UNDERFLOW");
   }
   else if(start->next == NULL)
   {
       start = NULL;
       free(head);
       printf("\nnode deleted\n");
   }
   else
   {
       ptr = start;
       start = start -> next;
       start -> prev = NULL;
       free(ptr);
       printf("\nnode deleted\n");
   }

}
void deletion_last()
{
   struct node *ptr;
   if(start == NULL)
   {
       printf("\n UNDERFLOW");
   }
   else if(start->next == NULL)
   {
       start = NULL;
       free(start);
       printf("\nnode deleted\n");
   }
   else
   {
       ptr = start;
       if(ptr->next != NULL)
       {
           ptr = ptr -> next;
       }
       ptr -> prev -> next = NULL;
       free(ptr);
       printf("\nnode deleted\n");
   }
}
void display()
{
   struct node *ptr;
   printf("\n printing values...\n");
   ptr = start;
   while(ptr != NULL)
   {
       printf("%d\n",ptr->data);
       ptr=ptr->next;
   }
}
```