Program 9.

```
# include <stdio.h>
# include <malloc.h>
struct node {
        struct node * prev;
        int data;
        struct node * next ;
    };
struct node* start = NULL;
struct node * create (struct node*);
struct node * display (struct node *);
struct node * insert_bef (struct node *);
struct node * del (struct node*);
int main() {
        int opt;
        do {
          printf ("\n Choose an option:");
          printf (" \n1. Create a list");
          printf (" \n2. Display");
          printf (" \n3. Add a node to the left");
          printf ("\n4. Delete a specific node");
           printf("\n 5. EXIT");
           printf ("\n Enter your case:");
           scanf ("%d", &opt);
           switch (opt) {
              case 1: start = create(start);
              printf("Doubly linked list created");
                break;
             case 2: start = display (start);
                break;
        case 3: start = insert_before(start);
          break;
        case 4: start: del(start);
          break;
      }
    } while(opt != 5);
    return 0;
  }
struct node * create (struct node * start) {
        struct node * new_node, * ptr;
        int num;
         printf(" Enter -1 to end");
         prinf("\nEnter the data")
         scanf("%d, & num);
          while(num != -1) {
               if (start == NULL) {
               new_node = (struct node *) malloc (sizeof
                                  (struct node));
               new_node -> prev = NULL;
               new_node -> data = num;
               new_node -> next = NULL;
               start = new_node;
            }
          else {
               ptr = start;
               new_node = (struct node *) malloc (sizeof (struct
                                                   node));
        new_node -> data = num;
        while (ptr -> next != NULL) {
             ptr = ptr -> next;
        ptr -> next = new_node;
        new_node -> prev = ptr;
        new_node -> next = NULL;
        start = new_node;
     }
     else {
         ptr = start;
         new_node = (struct node*) malloc (sizeof (struct node));
         new_node -> data = num;
         while (ptr -> next != NULL)
                 ptr = ptr -> next;
         ptr -> next = new_node;
         new_node -> prev = ptr;
         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 the data :");
   scanf ("%d",&num) ;
   new_node=(struct node *) malloc (sizeof( struct node));
    new_node -> data = num;
    start -> prev = new_node ;
     new_node -> prev = NULL;
     new_node -> next = start ;
     start = new_node ;
     return start ;
  }
struct node * del ( struct node * start) {
   struct node * temp ;
      cint ele;
    if ( start = = NULL){
       printf ( "Empty list \n");
       return;
    }
      printf("Enter the element to be deleted \n");
      scanf ("%d" &del;
      temp = start;
      while ( temp -> data != ele) {
         temp = temp -> next ;
           if ( temp = = NULL) {
              printf ("Element is not in the list \n");
            break;
           }
      }
    if ( temp = = start ) {
           start = start -> next;
      }
else if (temp -> next = = NULL) {
            temp = temp -> prev;
            temp -> next = NULL;
  }
else {
        temp -> prev -> next = temp -> next;
        temp -> next -> prev = temp -> prev;
   }
 return start;
}.
```