Collage : Vishwakarma Institute of Technology
Course Name : Data Structure in C
Name : Vedika Vikas Sontakke
Roll no : 37
PRN NO 12220206

Assignment No 2 : Implement following Assignments Based on Linked list..

a. Create Single Linked list and implement insert , delete , display operation.
   Program :

```c
#include <stdio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *next;
};

struct node *head , *temp = NULL;
void display();
void delete_specific_node()
{
    struct node* prev = head;
    struct node* current = head;
    int y;
    printf("enter the element which you want to delete :\n");
    scanf("%d" , &y);

    while(current->data != y)
    {
       prev = current;
       current = current->next;
    }

    prev->next = current->next;
    current->next = NULL;
    free(current);
```

```
    printf("%d element deleted successfully \n" , y);
    display();
}
void delete_last_node()
{
    struct node* prev = head;
    struct node* current = head;
    while(current->next != NULL){
       prev = current;
       current = current->next;
    }
    prev->next = NULL;
    free(current);
    printf("last element deleted successfully \n");
    display();

    delete_specific_node();
}

void delete_first_node()
{
   struct node* first = head;
   head = head->next;
   free(first);
   printf("first element deleted successfully \n");
   display();

   delete_last_node();
}

void insert_before_specific_node()
{
    struct node* current = head;
    struct node* prev = head;

   struct node *newnode = (struct node *)malloc(sizeof(struct node *));
   int y;
   printf("enter element before that you want to insert at element : \n");
   scanf("%d" , &y);

   printf("enter the ellement which you want to insert :\n");
   scanf("%d" , &newnode->data);

   newnode->next = NULL;
```

```
   while(current->data != y){
       prev = current;
      current = current->next;
   }

   newnode->next = current;
   prev->next = newnode;

   display();

   delete_first_node();
}

void insert_after_specific_node()
{
   struct node* current = head;
   struct node *newnode = (struct node *)malloc(sizeof(struct node *));
   int y;
   printf("enter element after that you want to insert at element : \n");
   scanf("%d" , &y);

   printf("enter the ellement which you want to insert :\n");
      scanf("%d" , &newnode->data);

   newnode->next = NULL;

   while(current->data != y)
      current = current->next;

   newnode->next = current->next;
   current->next = newnode;

   display();
   insert_before_specific_node();
}

void insert_end()
{
   struct node *newnode = (struct node *)malloc(sizeof(struct node *));

   printf("enter element which you want to insert at end : \n");
   scanf("%d" , &newnode->data);
   newnode->next = NULL;
   struct node *current_ = head;
```

```
   while( current_->next != NULL )
        current_ = current_->next;

   current_->next = newnode;

   display();
   insert_after_specific_node();
}

void addnode(int x)
{
    struct node *newnode = (struct node*)malloc(sizeof(struct node));
    newnode->data = x;
    newnode->next = NULL;

    if(head == NULL)
    {
        head = newnode;
        temp = newnode;
    }else
    {
        temp->next = newnode;
        temp = newnode;
    }
}
void insert_begin()
{
    struct node *newnode = (struct node *)malloc(sizeof(struct node *));

   printf("enter element which you want to insert at begin : \n");
   scanf("%d" , &newnode->data);

    newnode->next = head;
    head = newnode;

   display();
   insert_end();
}

void display()
{
    struct node *current = head;

    printf("elements in the linked list are :");
```

```
     while(current != NULL)
     {
        printf("%d ", current->data);
        current = current->next;
     }

     printf("\n");
}

void creation()
{
    int size;
    printf("enter the size of linked list :");
    scanf("%d",&size);
    printf("enter elements of linked list :\n");

    for(int i=0 ; i<size ; i++)
    {
       int n;
       scanf("%d",&n);
       addnode(n);
    }
    display();
    insert_begin();
}

int main()
{
    creation();
}
```

Output :

```
PS C:\Users\Lenovo\Documents\vit\data structure in c> cd "c:\Users\Lenovo\Documents\vit\data structure in c\linked list\" ; if ($?) { gcc linked_list_create_inse
rt_delete.c -o linked_list_create_insert_delete } ; if ($?) { .\linked_list_create_insert_delete }
enter the size of linked list :4
enter elements of linked list :
1 2 3
4
elements in the linked list are :1 2 3 4
enter element which you want to insert at begin :
22
elements in the linked list are :22 1 2 3 4
enter element which you want to insert at end :
33
elements in the linked list are :22 1 2 3 4 33
enter element after that you want to insert at element :
1
enter the ellement which you want to insert :
44
elements in the linked list are :22 1 44 2 3 4 33
enter element before that you want to insert at element :
2
enter the ellement which you want to insert :
55
elements in the linked list are :22 1 44 55 2 3 4 33
first element deleted successfully
elements in the linked list are :1 44 55 2 3 4 33
last element deleted successfully
elements in the linked list are :1 44 55 2 3 4
enter the element which you want to delete :
2
2 element deleted successfully
elements in the linked list are :1 44 55 3 4
PS C:\Users\Lenovo\Documents\vit\data structure in c\linked list>
```

b. Create Doubley linked list and implement insert , delete , display operation.

Program :

```c
#include<stdio.h>
#include<stdlib.h>
struct node
{
    struct node *prev;
    int data;
    struct node *next;
};
struct node *head , *temp = NULL;

void display()
{
   struct node *temp = head;

    printf("elements in the linked list are :");
    while(temp != NULL){

        printf("%d " ,temp->data);
        temp = temp->next;
    }
```

```
    printf("\n");
}

void delete_speciefic_node(){
     int y ;
     struct node *temp = head;

     printf("enter element which you want delete \n");
     scanf("%d",&y);

    while(temp->data != y)
       temp = temp->next;

    temp->prev->next = temp->next;
    temp->next->prev = temp->prev;

    display();
}

void delete_last_node()
{
    struct node *prev = head;
    struct node *first = head;
    while(first->next != NULL){
        prev = first;
        first = first->next;
    }
    prev->next = NULL;
    free(first);
    printf("last element deleted successfully \n");
    display();
}
void delete_first_node()
{
    struct node *first = head;
    head = head->next;
    free(first);
    printf("first element deleted successfully \n");
    display();

}
void insert_element_before_specefic_node()
{
   struct node *temp = head;
   struct node *newnode_ = (struct node *)malloc(sizeof(struct node *));
```

```
   int y;
   printf("enter element before that you want to insert at element : \n");
   scanf("%d" , &y);

   printf("enter the element which you want to insert :\n");
   scanf("%d" , &newnode_->data);

   while(temp->data != y)
        temp = temp->next;

    newnode_->prev = temp->prev;
    temp->prev->next = newnode_;
    temp->prev = newnode_;
    newnode_->next = temp;

   display();
}

void insert_after_specific_node()
{
    struct node *temp = head;
   struct node *newnode = (struct node *)malloc(sizeof(struct node *));
   int y;
   printf("enter element after that you want to insert at element : \n");
   scanf("%d" , &y);

   printf("enter the ellement which you want to insert :\n");
   scanf("%d" , &newnode->data);

    while(temp->data != y)
      temp = temp->next;

     temp->next->prev = newnode;
     newnode->next = temp->next;
     temp->next = newnode;
     newnode->prev = temp;

   display();
}
void insert_end()
{
    struct node *temp = head;

    struct node *newnode = (struct node *)malloc(sizeof(struct node *));
```

```
   printf("enter element which you want to insert at end : \n");
   scanf("%d" , &newnode->data);
   newnode->next = NULL;

   while(temp->next != NULL)
     temp = temp->next;

   temp->next = newnode;
   newnode->prev = temp;

   display();
}
void insert_begin()
{
    struct node *newnode = (struct node *)malloc(sizeof(struct node));
    printf("enter element which you want to insert at begin : \n");
    scanf("%d" , &newnode->data);

    newnode->next = head;
    head = newnode;

    display();
}
void addnode(int x)
{
    struct node *newnode = (struct node*)malloc(sizeof(struct node));
    newnode->data = x;
    newnode->prev = NULL;
    newnode->next = NULL;

   if(head == NULL)
   {
     head = newnode;
     temp = newnode;
   }else
   {
     temp->next = newnode;
     newnode->prev = temp;
     temp = newnode;
   }
}
void creation()
{
    int size;
    printf("enter the size of linked list :");
```

```
    scanf("%d",&size);
    printf("enter elements of linked list :\n");

    for(int i=0 ; i<size ; i++)
    {
       int n;
       scanf("%d",&n);
       addnode(n);
    }

    display();

}


int main()
{
    creation();
    insert_begin();
    insert_end();
  //insert_after_specific_node();
  //insert_element_before_specefic_node();
    delete_first_node();
    delete_last_node();
    delete_speciefic_node();
}
```

Output :

```
PS C:\Users\Lenovo\Documents\vit\data structure in c> cd "c:\Users\Lenovo\Documents\vit\data structure in c\linked list\doubley linked list\" ; if ($?) { gcc cre
ate_insert_delete.c -o create_insert_delete } ; if ($?) { .\create_insert_delete }
enter the size of linked list :3
enter elements of linked list :
1
2
3
elements in the linked list are :1 2 3
enter element which you want to insert at begin :
20
elements in the linked list are :20 1 2 3
enter element which you want to insert at end :
30
elements in the linked list are :20 1 2 3 30
first element deleted successfully
elements in the linked list are :1 2 3 30
last element deleted successfully
elements in the linked list are :1 2 3
enter element which you want delete
2
elements in the linked list are :1 3
PS C:\Users\Lenovo\Documents\vit\data structure in c\linked list\doubley linked list>
```

c. Create Circular linked list and implement insert , delete , display operation.

Program :

```c
#include<stdio.h>
#include<stdlib.h>

struct node
{
    int data;
    struct node *next;
};

struct node *head , *temp = NULL;
void display()
{
    struct node *temp = head;
    printf("elements in the linked list are :");

    while(temp->next != head)
    {
        printf("%d ",temp->data);
        temp = temp->next;
    }
    printf("%d ",temp->data);
    printf("\n");
}

void delete_first_node()
{
    struct node *temp = head;
    while(temp->next != head)
      temp = temp->next;

    head = head->next;
    temp->next = head;

    printf("first element deleted successfully \n");
    display();
}

void delete_last_node()
```

```
{
    struct node *temp = head;
    struct node *prev = head;
    while(temp->next != head){
        prev = temp;
        temp = temp->next;
    }

   prev->next = head;
   printf("last node deleted successfully \n");
   display();
}

void delete_specefic_node()
{
   struct node* prev = head;
    struct node* current = head;
    int y;
    printf("enter the element which you want to delete :\n");
    scanf("%d" , &y);

    while(current->data != y)
    {
       prev = current;
       current = current->next;
    }

    prev->next = current->next;
    current->next = NULL;
    free(current);

    printf("%d element deleted successfully \n" , y);
    display();
}

void insert_begin()
{
   struct node *newnode = (struct  node*)malloc(sizeof(struct node ));
   printf("enter element which you want to insert at begin : \n");
   scanf("%d" , &newnode->data);
   newnode->next = NULL;
   struct node *temp = head;

   newnode->next = head;
```

```
   while(temp->next != head)
     temp = temp->next;

   temp->next = newnode;
   head = newnode;

   display();

}

void insert_end()
{
   struct node *temp = head;
   struct node *newnode = (struct node *)malloc(sizeof(struct node *));

   printf("enter element which you want to insert at end : \n");
   scanf("%d" , &newnode->data);

   while(temp->next != head)
      temp = temp->next;

    temp->next = newnode;
    newnode->next = head;

    display();
}

void insert_before_specific_node()
{
    struct node* temp = head;
    struct node* prev = head;

   struct node *newnode = (struct node *)malloc(sizeof(struct node *));
   int y;
   printf("enter element before that you want to insert at element : \n");
   scanf("%d" , &y);

   printf("enter the ellement which you want to insert :\n");
   scanf("%d" , &newnode->data);

   newnode->next = NULL;

   while(temp->data != y){
       prev = temp;
      temp = temp->next;
```

```
   }

   newnode->next = temp;
   prev->next = newnode;

   display();
}

void insert_after_specific_node()
{
   struct node* temp = head;
   struct node *newnode = (struct node *)malloc(sizeof(struct node *));
   int y;
   printf("enter element after that you want to insert at element : \n");
   scanf("%d" , &y);

   printf("enter the ellement which you want to insert :\n");
   scanf("%d" , &newnode->data);

   newnode->next = NULL;

   while(temp->data != y)
      temp = temp->next;

   newnode->next = temp->next;
   temp->next = newnode;

   display();
}

void addnode(int n)
{
   struct node *newnode = (struct  node*)malloc(sizeof(struct node ));
   newnode->data = n;
   newnode->next = NULL;
   if(head == NULL)
   {
      head = newnode;
      temp = newnode;
      newnode->next = head;
   }else
   {
     newnode->next = head;
     temp->next = newnode;
     temp = newnode;
```

```
   }
}
int main()
{
    int size;
   printf("enter the size of list :");
   scanf("%d",&size);

   printf("enter the elements in the list \n");

   for(int i=0 ; i<size ; i++)
   {
      int n;
      scanf("%d",&n);
      addnode(n);
   }

   display();
   insert_begin();
   insert_end();
   insert_after_specific_node();
   insert_before_specific_node();
   delete_first_node();
   delete_last_node();
  delete_specefic_node();

}
```

Output :

```
enter the size of list :3
enter the elements in the list
1 2 3
elements in the linked list are :1 2 3
enter element which you want to insert at begin :
10
elements in the linked list are :10 1 2 3
enter element which you want to insert at end :
20
elements in the linked list are :10 1 2 3 20
enter element after that you want to insert at element :
2
enter the ellement which you want to insert :
55
elements in the linked list are :10 1 2 55 3 20
enter element before that you want to insert at element :
2
enter the ellement which you want to insert :
88
elements in the linked list are :10 1 88 2 55 3 20
first element deleted successfully
elements in the linked list are :1 88 2 55 3 20
last node deleted successfully
elements in the linked list are :1 88 2 55 3
enter the element which you want to delete :
2
2 element deleted successfully
elements in the linked list are :1 88 55 3
PS C:\Users\Lenovo\Documents\vit\data structure in c\linked list\circular linked list>
```