Date:- 22/1/2024

① implement singly linked list with following operation.

ⓐ create a linked list
ⓑ insertion of a node at first position at any position & at the end position.

singly linked list.

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

struct node {
    struct node *data;
    struct node *next;
};
struct node *head = NULL;
struct node *create_node();
struct node *insert_beg();
struct node *insert_end();
struct node *insert_pos();
```

```
struct node *display();

int main()
{
    int option;
    while(1)
    {
        printf("\n1.Create node \n2. insert at begng.
            \n3. insert at end. \n4. Insert at random
            \n6. exit \n5. display")
        scanf("%d", &option);

        switch(option)
        {
            case 1: Create_node(*head);
                    Break;
            case 2: insert_beg(*head);
                    Break;
            case 3: insert_end(*head);
                    Break;
            case 4: insert_ran(*head);
                    Break;
            case 5: display();
                    Break;
            case 6: exit(0);
            default: printf("wrong value");
                    Break;
        }
    }
}
```

```
struct node *create_node(*head)
{
    struct node * newnode, *temp;
    temp = head;
    int n;
    printf("How many data nodes you want create")
    scanf("%d" &n);

    for(int i=1; i<n; i++)
    {
      newnod=(struct node*) malloc(sizeof(struct
                                        node));
      printf("Enter data");
      scanf("%d", &newnode->data);
      newnode = h
      newnode->next =null.

      if(head ==null)
      {
       temp=head=newnode;
      }
    else.
      {
         temp->next=new_node;
         temp=newnode;
      }
}
```

```
struct node *insert_beg(head)
{
 struct node *newnode, *temp;

 newnode=(struct node*)malloc(sizeof(struct node));
 printf("enter data");
 scanf("%d",&newnode->data);
  newnode->next = head;
  head = newnode;
 }
}

struct node *insert_end(head)
{
    struct node *newnode, *temp;
   newnode=(struct node*) malloc(sizeof(struct node));
   temp=head; newnode->next=NULL;
   while(temp->next!=NULL)
     {
        temp=temp->next)
     }
     temp->next = new node;
   }

struct node *insert_ran(head)
{
    int pos;
   struct node *newnode, *temp; *prenode

 printf("Enter the position");
 scanf("%d",&pos);
 temp = head;
 newnode=(struct node*) malloc(sizeof(struct node)
        );
```

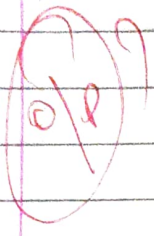
```
printf("enter the data:");
scanf("%d", &newnode->data);
int i=1
while(i<pos)
{
    temp=temp->next;
    i++;
}
newnode->next=temp->next;
;
temp->next=newnode;
}
```

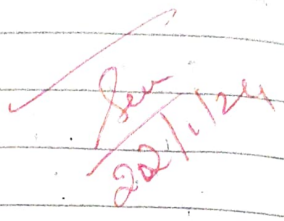
2.

```
struct node Delete_Beg(head)
{
    struct node *head, *temp;

if(head==null)
{
  printf("empty");
}

else
{
  temp=head;
  head=temp->next;
  free(temp);
}
}
```

```
struct node *delete_end(head);
{
    struct node *prenode, *temp;
    temp = head;
    if(head == NULL)
    {
        printf("empty");
        free(head);
    }
    else
    {
        while(temp->next != NULL)
        {
            prenode = temp;
            temp = temp->next;
        }
        free(temp);
        prenode->next = NULL;
    }

struct node *delete_ran(head)
{
    struct node *temp, *next node;
    temp = head;
    int pos, i = 1;
    printf("enter position");
    scanf("%d", &pos);
    while(i<pos-1)
    {
        temp = temp->next;
        i++;
    }
    nextnode = temp->next;
```

temp->next = nextnode->next;
free(nextnode);
}





O/P

① create node
② insert at beging
③ insert at end
④ insert at random position.
⑤ display
⑥ delete at begining
⑦ delete at end
⑧ delete at random position
⑨ exit.

Enter the option ①
how many data you want to insert 3
Enter data 23
Enter data 45
Enter data 76

Enter the option 5
23
45
76

Enter the option 2
Enter data 78
Enter the option 5

Date __/__/__
Page _____

78 23
23
45
76

enter the option 6
deleted element is 78.

enter the option 7
deleted element is 76

enter the option 8
enter position 2
deleted element is 23

enter options 5
45

5