NAME:- Aindrail Santra

RollNo :-2029044

Create a single linked list and perform following operations:

1. (a) Insert the new list at any position.
(b) Insert the new list in front.
(c) Insert the new list at the end.

```
#include<stdio.h>
#include<stdlib.h>
struct Node{
int data;
struct Node * next;
};
struct Node * insert(int x)
{
struct Node *head=NULL;
struct Node *temp=NULL;
struct Node *p=NULL;
for(int i=0; i<x  i++); {
temp = (struct Node *) malloc(sizeof(struct Node));
printf("Enter No.  "); :
scanf("%d",&(temp->data));
temp->next=NULL;
```

NAME:- Aindrail Santra

RollNo :-2029044

```
if(head==NULL)
{
head=temp;
                    }
else{
p=head;
while(p->next!=NULL)
p=p->next;
p->next=temp;
}}
return head;
}

//a
struct tNode * insertAtFirst(struct tNode *head, int data){
struct tNode * ptr= (struct tNode *) malloc(sizeof(struct tNode));
ptr->data = data;
ptr->next= head;
return ptr;
}
//b
struct tNode * insertAtIndex(struct tNode *head, int data, int
index){
```

NAME:- Aindrail Santra

RollNo :-2029044

```
structNode * ptr= (structNode *) malloc(sizeof(structNode));
structNode * p = head;
inti = 0;
while (i!=index-1)
{
p = p-> next;
i++;
        }
ptr-> data = data;
ptr-> next= p-> next;
p-> next= ptr;
return head;
}
//c
structNode * insertAtEnd(structNode *head, intdata){
structNode * ptr= (structNode *) malloc(sizeof(structNode));
ptr-> data = data;
structNode * p = head;
while(p-> next!=NULL){
p = p-> next;
        }
p-> next= ptr;
```

NAME:- Aindrail Santra
RollNo :-2029044

```
ptr->next=NULL;
return head;
}
void linkedListTraversal(structNode *ptr)
{
while(ptr =NULL)
{
printf(" %d,", ptr->data);
ptr= ptr->next;
        }
}
intmain()
{  ntn,x; i
printf("NO. ofNodesinLinkedList "); :
scanf("%d",& n);
structNode *HEAD=NULL;
HEAD = insert(n);
printf("*LINKEDLIST**\n");
linkedListTraversal(HEAD);
printf("\n***AFTERINSERTION**\n");
HEAD = insertAtEnd(HEAD, 69);
linkedListTraversal(HEAD);
return 0;
}
```

NAME:- Aindrail Santra

RollNo :-2029044

OUTPUT

```
NO. ofNodesinLinkedList 7  :
EnterNo. 21:
EnterNo. 34:
EnterNo. 221
EnterNo. 23:
EnterNo. 2 :
EnterNo. 12:
EnterNo. 12:
*LINKEDLIST**
21, 34, 221, 23, 2, 12, 12,
***AFTERINSERTION**
21, 34, 221, 23, 2, 12, 12, 69,
```

NAME:- Aindrail Santra
Roll No :- 2029044

Create a single linked list and perform following operations:
2. (a) Delete the list from any postion.
(b) Delete the list from the front.
(c) Delete the list from the end.

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
int data;
struct node *next;
};
void create(struct node **head)
{
struct node *current, *ptr;
for(int i = 0; i < 8; i++)
{
current = malloc(sizeof(struct node));
current->data = rand() % 100 + 1;
current->next = NULL;
if (*head == NULL)
{
*head = current;
```

NAME:- Aindrail Santra
Roll No :- 2029044

```
ptr = current;
}
else
{
ptr->next = current;
ptr = current;
}}}
void insert(struct node **head,int val,int
pos){
struct node *temp=malloc(sizeof(struct
node));
temp->data=val;
temp->next=NULL;
if(*head==NULL || pos==0){
temp->next=*head;
*head=temp;
}
else{
struct node *ptr;
ptr=*head;
int i=1;
while(inext!=NULL){
i++;
ptr=ptr->next;
}
```

NAME:- Aindrail Santra
Roll No :- 2029044

```
temp->next=ptr->next;
ptr->next=temp;
}}
void delete (struct node **head, int val,int
pos)
{
if (*head == NULL)
printf("List empty");
else
{
struct node *ptr,*prev;
ptr = *head;
while (ptr != NULL)
{
if (ptr->data == val)
break;
prev = ptr;
ptr = ptr->next;
}
if (ptr == NULL)
printf("Data not found ");
else if (ptr == *head)
{
*head = ptr->next;
free(ptr);
```

NAME:- Aindrail Santra
Roll No :- 2029044

```
}
else
{
prev->next = ptr->next;
free(ptr);
}}}
void display(struct node *head)
{
struct node *current = head;
for(; current != NULL; current = current->next)
{
printf("%d ", current->data);
}
printf("\n");
}
int main()
{
int val;
struct node *head = NULL;
create(&head);
insert(&head,34,10);
insert(&head,67,59);
insert(&head,89,21);
display(head);
```

NAME:- Aindrail Santra
Roll No :- 2029044

```
delete (&head,34,10);
delete(&head,89,21);
delete(&head,67,59);
display(head);
return 0;
}
```

OUTPUT

```
42 68 35 1 70 25 79 59 34 67 89
42 68 35 1 70 25 79 59
```

NAME:- Aindrail Santra
Roll No :- 2029044

Q 3. WAP to search an element in a simple linked list, if found delete that node and insert that node at beginning. Otherwise display an appropriate message.

```
#include<stdio.h>
#include<stdlib.h>
struct node{
int data;
struct node *next;
}
*head = NULL;
tail = NULL;
void create (struct node *head, int n){
for(int i=0; i struct node *cur;
cur=(struct node*) malloc (sizeof(struct
node));
cur -> data = rand()%100;
cur->next = NULL;
if(head == NULL)
{
head = tail = cur;
}
else{
```

NAME:- Aindrail Santra
Roll No :- 2029044

```
tail->next=cur;
tail=cur;
}}}
void insert(struct node **head, int n, int a)
{
struct node *cur;
struct node *ptr= &head;
cur=(struct node*)malloc(sizeof(struct
node));
cur->data=n;
cur->next=NULL;
if(*head == NULL )
{
cur-> next = *head;
*head = cur;
}
else{
int i=1;
while (ptr!=NULL && i {
i++;
ptr = ptr->next;
}
if(ptr != NULL){
cur -> next = ptr -> next;
```

NAME:- Aindrail Santra
Roll No :- 2029044

```
ptr->next = cur;
}}}
void delete (struct node **head, int del)
{
struct node *qtr;
struct node *ptr = *head ;
int i=0;
if (*head == NULL){
printf("The list is empty");
}
if (del == (*head)-> data)
{
qtr = ptr-> next;
*head=qtr;
free(ptr);
}
else{
while(ptr-> next){
if(ptr-> data != del){
{
if (ptr -> next -> data == del || ptr -> next
== NULL){
break;
}}
```

NAME:- Aindrail Santra
Roll No :- 2029044

```
ptr = ptr -> next;
}
qtr = ptr-> next;
ptr -> next = qtr-> next;
free (ptr);
}}
void display (struct node *p)
{
if (p==NULL){
printf("\n The list is empty");
}
printf("\n");
while (p == NULL){
printf("%d", p-> data);
p = p-> next;
}}
int main(){
int n,d,i;
printf("Enter the length of the list");
scanf("%d, & n");
create(& head, n);
printf("list before insertion");
display(head);
printf("\n Enter the data to be
```

NAME:- Aindrail Santra
Roll No :- 2029044

```
deleted:");
scanf("%d", &d);
printf("List after deletion:");
delete(&head, d,0);
display(head);
return 0;
}
```

OUTPUT

```
Enter the length of the list: 5
List before insertion :
88 86 78 15 92
Enter the data to be deleted : 78
List after deletion :
88 85 15 92
List after insertion:
78 88 86 15 92
```

NAME:- Aindrail Santra
Roll No :- 2029044

Q 4. WAP to check whether a singly linked list is a palindrome or not.

```
#include<stdio.h>
#include<stdbool.h>
#include<stdlib.h>
struct node{
int data;
struct node *next;
};
struct node *head, *tail = NULL;
int size = 0;
void addNode(int data) {
struct node *newNode = (struct
node*)malloc(sizeof(struct node));
newNode->data = data;
newNode->next = NULL;
if(head == NULL) {
head = newNode;
tail = newNode;
}
else {
tail->next = newNode;
tail = newNode;
}
```

NAME:- Aindrail Santra
Roll No :- 2029044

```
size++;
}
struct node* reverseList(struct node *temp){
struct node *current = temp;
struct node *prevNode = NULL, *nextNode =
NULL;
while(current != NULL){
nextNode = current->next;
current->next = prevNode;
prevNode = current;
current = nextNode;
}
return prevNode;
}
void isPalindrome(){
struct node *current = head;
bool flag = true;
int mid = (size%2 == 0)? (size/2) :
((size+1)/2);
for(int i=1; i current = current->next;
}
struct node *revHead =
```

NAME:- Aindrail Santra
Roll No :- 2029044

```
reverseList(current->next);
while(head != NULL && revHead != NULL){
if(head->data != revHead->data){
flag = false;
break;
}
head = head->next;
revHead = revHead->next;
}
if(flag)
printf("Given singly linked list is a
palindrome\n");
else
printf("Given singly linked list is not a
palindrome\n");
}
void display() {
struct node *current = head;
if(head == NULL) {
printf("List is empty\n");
return;
}
printf("Nodes of singly linked list: \n");
```

NAME:- Aindrail Santra
Roll No :- 2029044

```
while(current != NULL) {
printf("%d ", current->data);
current = current->next;
}
printf("\n");
}
int main()
{
addNode(9);
addNode(5);
addNode(3);
addNode(5);
addNode(9);
display();
isPalindrome();
return 0;
}
```

OUTPUT

```
Nodes of singly linked list:
9 5 3 5 9
Given singly linked list is a palindrome
```

NAME:- Aindrail Santra
Roll No :- 2029044

Q 5. WAP to display the contents of a linked list in reverse order.

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
int data;
struct node *next;
}*head=NULL,*last=NULL;
int count();
void create(int key)
{
struct node *t;
t=(struct node *)malloc(sizeof(struct node));
t->data=key;
t->next=NULL;
if(head==NULL)
head=last=t;
else
{
t->next=last;
last=t;
}}
void display(struct node *p)
```

NAME:- Aindrail Santra
Roll No :- 2029044

```
{
printf("\n");
while(p)
{
printf("%d ",p->data);
p=p->next;
}}
void release(void)
{
struct node *temp1=head,*temp2;
while(temp1)
{
temp2=temp1->next;
free(temp1);
temp1=temp2;
}}
void main()
{
create(1);
create(2);
create(9);
create(6);
display(last);
```

NAME:- Aindrail Santra
Roll No :- 2029044

```
release();}
```

OUTPUT

1 2 9 6
6 9 2 1

Q 6. Given a singly linked list, rotate the linked list counter-clockwise by k nodes. Where k is a given positive integer.

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
int info;
struct node * next;
};
struct node * start = NULL;
void add(int item)
{
struct node * t, * p;
t = (struct node *)malloc( sizeof( struct node
));
if( start == NULL)
{
```

NAME:- Aindrail Santra
Roll No :- 2029044

```
start = t;
start->info = item;
start->next = NULL;
return;
}
else
{
struct node * p = start;
while(p->next != NULL)
{
p = p->next;
}
p->next = t;
p=p->next;
p->info = item;
p->next = NULL;
}}

void rotate(struct node * head, int k)
{
if (k == 0)
return;
struct node* curr = head;
int count = 1;
```

NAME:- Aindrail Santra
Roll No :- 2029044

```
while (count < k && curr != NULL) {
curr = curr -> next;
count++;
}
if (curr == NULL)
return;
struct node* temp = curr;
while (curr -> next != NULL)
curr = curr -> next;
curr -> next = head;
head = temp -> next;
temp -> next = NULL;
start = head;
}
void traverse(struct node * t)
{
if(t == NULL)
{
printf(" Linked list is empty\n");
}
while(t -> next != NULL)
{
printf("%d -> ",t -> info);
t = t -> next;
}
```

NAME:- Aindrail Santra
Roll No :- 2029044

```
printf("%d\n",t -> info);
}
int main()
{
int k;
printf("Enter the value of k:");
scanf("%d",&k);
add(4);
add(8);
add(12);
add(16);
add(20);
add(24);
add(28);
add(32);
printf("Given Linked List: \n");
traverse(start);
rotate(start,k);
printf("Linked List after rotating: \n");
traverse(start);
return 0;}
```

OUTPUT

```
Enter the value of k:3
```

NAME:- Aindrail Santra
Roll No :- 2029044

Given Linked List:
4 -> 8 -> 12 -> 16 -> 20 -> 24 -> 28 -> 32
Linked List after rotating:
16 -> 20 -> 24 -> 28 -> 32 -> 4 -> 8 -> 12

Q 7. WAP to remove duplicates from a linked list of n nodes.

```
#include<stdio.h>
#include<stdlib.h>
struct Node
{
int data;
struct Node *next;
}*first=NULL,*second=NULL,*third=NULL;
void Display(struct Node *p)
{
while(p!=NULL)
{
printf("%d ",p->data);
p=p->next;
}}
```

NAME:- Aindrail Santra
Roll No :- 2029044

```
void create(int A[],int n)
{
int i;
struct Node *t,*last;
first=(struct Node *)malloc(sizeof(struct
Node));
first->data=A[0];
first->next=NULL;
last=first;
for(i=1;i<n;i++) {
t=(struct Node*)malloc(sizeof(struct Node));
t->data=A[i];
t->next=NULL;
last->next=t;
last=t;
}
}
void RemoveDuplicate(structNode *p)
{
struct Node *q=p->next;
while(q!=NULL)
{
if(p->data!=q->data)
```

NAME:- Aindrail Santra
Roll No :- 2029044

```
{
p=q;
q=q->next;
}
else
{
p->next=q->next;
free(q);
q=p->next;
}}}
int main()
{
int A[]={10,30,30,30,50,50,50,60};
create(A,8);
RemoveDuplicate(first);
Display(first);
return 0;
}
```

OUTPUT

10 30 30 30 50 50 50 60

10 30 50 60