

## LAB - 5 & 6

/\*C-Program to implement LINKED LIST with functions insertion & deletion with specified position

LAB 5 & 6

\*/

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
    int info;
    struct node *link;
};
typedef struct node *NODE;
NODE getnode(){
    NODE x;
    x=(NODE)malloc(sizeof(struct node));
    if(x==NULL){
        printf("mem full\n");
        exit(0);
    }
    return x;
}
void freenode(NODE x){
    free(x);
}
NODE insert_rear(NODE first,int item){
    NODE temp,cur;
    temp=getnode();
    temp->info=item;
    temp->link=NULL;
    if(first==NULL)
        return temp;
    cur=first;
    while(cur->link!=NULL)
        cur=cur->link;
    cur->link=temp;
    return first;
}
NODE delete_rear(NODE first){
    NODE cur,prev;
    if(first==NULL){
```

```

    printf("list is empty cannot delete\n");
    return first;
}
if(first->link==NULL){
    printf("item deleted is %d\n",first->info);
    free(first);
    return NULL;
}
prev=NULL;
cur=first;
while(cur->link!=NULL){
    prev=cur;
    cur=cur->link;
}
printf("item deleted at rear-end is %d",cur->info);
free(cur);
prev->link=NULL;
return first;
}
NODE insert_pos(int item,int pos,NODE first){
    NODE temp,cur,prev;
    int count;
    temp=getnode();
    temp->info=item;
    temp->link=NULL;
    if(first==NULL&&pos==1){
        return temp;
    }
    if(first==NULL){
        printf("invalid position\n");
        return first;
    }
    if(pos==1){
        temp->link=first;
        first=temp;
        return temp;
    }
    count=1;
    prev=NULL;
    cur=first;
    while(cur!=NULL&&count!=pos){
        prev=cur;
        cur=cur->link;
        count++;
    }

```

```

}
if(
    count==pos){
    prev->link=temp;
    temp->link=cur;
    return first;
}
printf("invalid position\n");
return first;
}
NODE delete_pos(int pos,NODE first){
    NODE cur;
    NODE prev;
    int count,flag=0;
    if(first==NULL || pos<0){
        printf("invalid position\n");
        return NULL;
    }
    if(pos==1){
        cur=first;
        first=first->link;
        freenode(cur);
        return first;
    }
    prev=NULL;
    cur=first;
    count=1;
    while(cur!=NULL){
        if(count==pos){
            flag=1;
            break;
        }
        count++;
        prev=cur;
        cur=cur->link;
    }
    if(flag==0){
        printf("invalid position\n");
        return first;
    }
    printf("item deleted at given position is %d\n",cur->info);
    prev->link=cur->link;
    freenode(cur);
    return first;
}

```

```

}
void display(NODE first){
    NODE temp;
    if(first==NULL)
        printf("list empty cannot display items\n");
    for(temp=first;temp!=NULL;temp=temp->link){
        printf("%d\n",temp->info);
    }
}
void main()
{
    int item,choice,key,pos;
    int count=0;
    NODE first=NULL;
    for(;;){

        printf("\n1:Insert_rear\n2:Delete_rear\n3:insert_info_position\n4:Delete_info_position\n5:Display
        _list\n6:Exit\n");
        printf("enter the choice\n");
        scanf("%d",&choice);
        switch(choice){
            case 1:printf("enter the item at rear-end\n");
                scanf("%d",&item);
                first=insert_rear(first,item);
                break;
            case 2:first=delete_rear(first);
                break;
            case 3:printf("enter the item to be inserted at given position\n");
                scanf("%d",&item);
                printf("enter the position\n");
                scanf("%d",&pos);
                first=insert_pos(item,pos,first);
                break;
            case 4:printf("enter the position\n");
                scanf("%d",&pos);
                first=delete_pos(pos,first);
                break;
            case 5:display(first);
                break;
            default:exit(0);
                break;
        }
    }
}

```

Command Prompt - linked-list1

D:\coding files\DS lab>linked-list1

```
1:Insert_rear
2:Delete_rear
3:insert_info_position
4:Delete_info_position
5:Display_list
6:Exit
enter the choice
1
enter the item at rear-end
12
```

```
1:Insert_rear
2:Delete_rear
3:insert_info_position
4:Delete_info_position
5:Display_list
6:Exit
enter the choice
1
enter the item at rear-end
13
```

```
1:Insert_rear
2:Delete_rear
3:insert_info_position
4:Delete_info_position
5:Display_list
6:Exit
enter the choice
1
enter the item at rear-end
45
```

```
1:Insert_rear
2:Delete_rear
3:insert_info_position
4:Delete_info_position
5:Display_list
6:Exit
enter the choice
1
```

Command Prompt - linked-list1

```
6:Exit
enter the choice
1
enter the item at rear-end
234

1:Insert_rear
2:Delete_rear
3:insert_info_position
4:Delete_info_position
5:Display_list
6:Exit
enter the choice
1
enter the item at rear-end
22

1:Insert_rear
2:Delete_rear
3:insert_info_position
4:Delete_info_position
5:Display_list
6:Exit
enter the choice
5
12
13
45
234
22

1:Insert_rear
2:Delete_rear
3:insert_info_position
4:Delete_info_position
5:Display_list
6:Exit
enter the choice
2
item deleted at rear-end is 22
1:Insert_rear
2:Delete_rear
3:insert_info_position
```

C:\ Command Prompt - linked-list1

2:Delete\_rear

3:insert\_info\_position

4:Delete\_info\_position

5:Display\_list

6:Exit

enter the choice

5

12

13

45

234

1:Insert\_rear

2:Delete\_rear

3:insert\_info\_position

4:Delete\_info\_position

5:Display\_list

6:Exit

enter the choice

3

enter the item to be inserted at given position

99

enter the position

4

1:Insert\_rear

2:Delete\_rear

3:insert\_info\_position

4:Delete\_info\_position

5:Display\_list

6:Exit

enter the choice

5

12

13

45

99

234

1:Insert\_rear

2:Delete\_rear

3:insert\_info\_position

4:Delete\_info\_position

Command Prompt - linked-list1

5  
12  
13  
45  
99  
234

1:Insert\_rear  
2>Delete\_rear  
3:insert\_info\_position  
4>Delete\_info\_position  
5:Display\_list  
6:Exit

enter the choice

4

enter the position

2

item deleted at given position is 13

1:Insert\_rear  
2>Delete\_rear  
3:insert\_info\_position  
4>Delete\_info\_position  
5:Display\_list  
6:Exit

enter the choice

5

12

45

99

234

1:Insert\_rear  
2>Delete\_rear  
3:insert\_info\_position  
4>Delete\_info\_position  
5:Display\_list  
6:Exit

enter the choice



