Name: S Skanda
USN: 1BM19CS137

Date: 14/12/2020

```
#include<stdio.h>
#include<stdlib.h>
#include<process.h>
struct node
{ int info;
  struct node *rlink, *llink;
};
typedef struct node *NODE;
NODE getnode()
{ NODE x;
  x = (NODE) malloc (sizeof(struct node));
  if (x == NULL)
  { printf("memory full\n");
    exit(0);
  }
  return x;
}
Node insert_rear (NODE head, int item)
{ NODE temp, cur;
  temp = getnode();
  temp->rlink = temp->llink = NULL
  temp->info = item;
  cur = head->llink;
  head->llink = temp
  temp->rlink = head;
  head->info = head->info+1;
  return head;
}
```

```
NODE insert_leftpos(int item, NODE head)
{ NODE temp, cur, prev;
  if (head->rlink == head)
  { printf("List is empty");
    return head;
  }
  cur = head->rlink;
  while (cur != head)
  { if (item == cur->info) break;
    cur = cur->rlink;
  }
  if (cur == head)
  { printf("key not found\n");
    return head;
  }
  prev = cur->llink
  printf("Enter towards left of %d : ", item
  temp = getnode();
  scanf("%d", &temp->info);
  prev->rlink = temp;
  temp->llink = prev;
  cur->llink = temp;
  temp->rlink = cur;
  return head;
}
Node delete_rear(NODE head)
{ NODE cur, prev;
  if(head->rlink == head)
  { printf("list is empty");
    return head;
  }
  cur = head->llink;
```

```
prev = cur->llink;
head -> llink = prev;
prev -> rlink = head;
printf ("The node deleted is %d \n", cur->info);
free (cur);
return head;
}
NODE delete all key (int item, NODE head)
{ NODE prev, cur, next;
  int count;
  if (head -> rlink == head)
  { printf ("list is empty");
    return head;
  }
count = 0;
cur = head -> rlink;
while (cur != head)
{ if (item != cur -> info)
    cur = cur -> rlink;
}
{count++;
 prev = cur -> llink;
 next = cur -> rlink;
 prev -> rlink = next;
 next -> llink = prev;
 free (cur);
 cur = next;
}
}
if (count == 0)
  printf ("key not found");
```

```
    else
    printf("key found at %d position and is
                deleted \n", count);
    return head;
}
void display (NODE head)
{ NODE temp;
  if (head -> rlink = head)
  { printf("list empty");
    return;
  }
  for (temp=head->rlink; temp!=head; temp=temp->rlink)
  printf("%d \n", temp->info);
}

void main()
{ int item, choice, key;
  NODE head;
  head = getnode();
  head->rlink = head->llink = head;
  for (;;)
  {printf("\n 1.insert 2.insert left 3.display 4.delete key
            5.delete 6.exit \n");
  printf("Enter your choice");
  scanf("%d", &choice);
  switch(choice)
  { case 1: printf("Enter the item :");
            scanf("%d", &item);
            head = insert_rear(head, item);
            break;
    case 2: printf("Enter the key item);
            scanf("%d", &item);
            head = insert_left(head, item);
            break;
```

```
case 3: display(head);
        break;
case 4: printf("Enter the key item:");
        scanf("%d",&item);
        head=delete_all_key(item,head);
        break;
case 5: head=delelete_rear(head);
        break;
default: exit(0);
        break;
    }
  }
}
```