Queue Operation

DEV KANWAR
IBM19CS044

```
#include<stdio.h>
#include<stdlib.h>
void create(int);
void display(int);
void reverse(int);
void concat();
struct node
{ int data
 struct node *next;
};
struct node *head1=NULL, *head2=NULL, *third;
 int main(int argc, char **argv)
{ int choice, ch, str;
 do {
 printf("\n1. create\n2. Display\n3. Reverse\n4.
      Concatinate\n5. Exit");
 printf("\n Enter your choice");
  scanf("%d",&choice);
  switch(choice)
{ case 1 : printf("Enter the list to be added to:\n");
          scanf("%d",&str);
          create(str); break;
 case 2 : printf("Enter the string to be displayed\n");
          scanf("%d",&str)
          display(str); break;
```

```
Case 3: printf("Enter the list to be reversed\n");
        scanf("%d",&str);
        reverse(str); break;
Case 4: concat();
        display(1);
        break; }}
while(choice!=5);
}
Void create(int str)
{ struct node *newnode, *temp, *head;
  if(str==1)
   head=head1;
  else
   head=head2;
   int item;
  new node = (struct node *) malloc(size of(struct node));
  printf("Enter node data:");
  scanf("%d",&item);
   if(head==NULL)
  { new node->next=NULL;
    head = new node;
     if(str==1)
   head1=head;
  else
  head2=head;
```

```
printf("Node created\n");
}
else
{temp = head;
while(temp -> next != NULL)
{ temp = temp -> next;
}
temp -> next = new node;
new node -> next = NULL;
printf("Node created\n"); }}
void concat()
{ struct node *temp1 = head1, *temp2 = head2;
while(temp1 -> next != NULL)
    temp = temp1 -> next;
    temp1 -> next = temp2;
}
void reverse(int str)
{ struct node *prev = NULL, *current, *next = NULL;
if(str == 1)
current = head1;
else if(str == 2)
current = head2;
while(current != NULL)
{ next = current -> next;
current -> next = prev;
```

```
prev = current;
current = next;
}
if (str == 1)
  head1 = prev;
else if (str == 2)
  head2 = prev;
  display(str);
}
void display (int str)
{ struct node *ptr = NULL;
  if (str == 1)
    ptr = head1;
  else if (str == 2)
    ptr = head2;
  if (ptr == NULL)
  { printf ("Nothing to print \n");
  }
  else
  { while (ptr != NULL)
    { printf ("%d", ptr -> data);
      ptr = ptr -> next;
    }
  }
}
```