WEEK 5

2(a) .Stack Implementation

#include <stdio.h>

struct node

{

int data;

struct node \*next;

};

struct node \*head=NULL;

void push()

{

int i,data;

printf("Enter the data :");

struct node \*last=head;

struct node \*new\_node;

new\_node=(struct node\*)malloc(sizeof(struct node));

scanf("%d",&data);

new\_node->data=data;

new\_node->next=NULL;

if(head==NULL)

{

head=new\_node;

}

else

{

while(last->next!=NULL)

{

last=last->next;

}

last->next=new\_node;

}

printf("Element pushed successfully\n");

}

void pop()

{

struct node\*ptr;

struct node\*ptr1;

if(head==NULL)

printf("List is empty\n");

else if(head->next==NULL)

free(head);

else

{

ptr=head;

while(ptr->next!=NULL)

{

ptr1=ptr;

ptr=ptr->next;

}

free(ptr);

ptr1->next=NULL;

printf("Element popped successfully\n");

}

}

void display()

{

struct node \*node=head;

if(head==NULL)

printf("List is empty\n");

else

{

printf("Stack :\n");

while(node!=NULL)

{

printf("%d -> ",node->data);

node=node->next;

}

printf("NULL\n");

}

void main()

{

int ch;

printf("Enter 1:pop \n 2:push \n 3:display \n 4:exit\n");

while(ch!=4)

{

printf("Enter the choice:");

scanf("%d",&ch);

switch(ch)

{

case 1: pop();

break;

case 2:push();

break;

case 3:display();

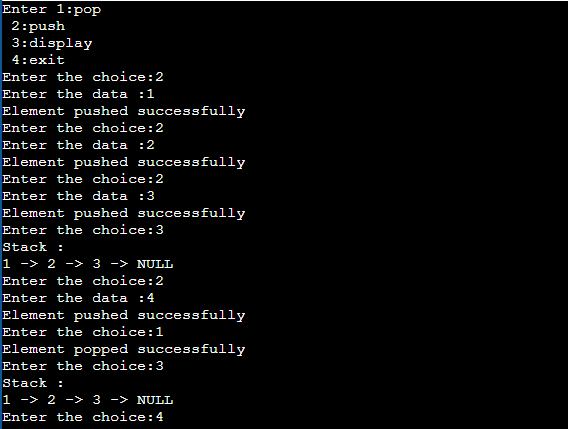
break;

}

}

}

OUTPUT:



2(B).Queue Implementation

#include <stdio.h>

struct node

{

int data;

struct node \*next;

};

struct node \*head=NULL;

void enqueue()

{

int i,data;

printf("Enter the data :");

struct node \*last=head;

struct node \*new\_node;

new\_node=(struct node\*)malloc(sizeof(struct node));

scanf("%d",&data);

new\_node->data=data;

new\_node->next=NULL;

if(head==NULL)

{

head=new\_node;

}

else

{

while(last->next!=NULL)

{

last=last->next;

}

last->next=new\_node;

}

}

void dequeue()

{

struct node \*ptr;

if(head == NULL)

printf("List is empty\n");

else

{

ptr=head;

head = ptr->next;

free(ptr);

printf("Node deleted from beginning\n");

}

}

void display()

{

struct node \*node=head;

if(head==NULL)

printf("List is empty\n");

else

{

printf("Queue:\n");

while(node!=NULL)

{

printf("%d -> ",node->data);

node=node->next;

}

printf("NULL\n");

}

}

void main()

{

int ch;

printf("Enter 1:Enqueue \n 2:Dequeue \n 3:Display \n 4:exit\n");

while(ch!=4)

{

printf("Enter the choice:");

scanf("%d",&ch);

switch(ch)

{

case 1:enqueue();

break;

case 2:dequeue();

break;

case 3:display();

break;

}

}

}

OUTPUT:

