**WEEK 7**

**Queue**

int main()

{

int ch; #include <stdio.h>

#include <stdlib.h>

struct Queue

{ int ele;

struct Queue \*next;};

typedef struct Queue q

q \*r=NULL;

;

q \*f=NULL;

void enqueue(int x)

{ q \*newnode=malloc(sizeof(q));

newnode->ele=x;

if(f==NULL && r==NULL)

{ f=r=newnode;

newnode->next=NULL;

return;

}

r->next=newnode;

r=newnode;

newnode->next=NULL;}

//f=newnode;}

void dequeue()

{ if(f==NULL && r==NULL)

{ printf("UNDERFLOW\n");

return;}

if(f==r)

{ printf("THE DELETED ELE IS %d\n",f->ele);

f=r=NULL;

return;}

q \*temp=f;

printf("DELETED ELEMENT IS %d\n",temp->ele);

f=f->next;

free(temp);

}

void display()

{ q \*temp=f;

while(temp!=NULL)

{ printf("%d ",temp->ele);

temp=temp->next;

}

printf("\n");

}

int main()

{

int ch;

printf("1 TO ENQUEUE\n2 TO DEQUEUE\n3 TO DISPLAY\n");

do

{ printf("ENTER YOUR CHOICE ");

scanf("%d",&ch);

switch(ch)

{ case 1:

int x;

printf("ELEMENT TO BE ADDED");

scanf("%d",&x);

enqueue(x);

break;

case 2:

dequeue();

break;

case 3:

display();

break;

default:

break;

} } while(ch<=3);

printf("THANK YOU");

}

#include <stdio.h>

#include <stdlib.h>

#define SIZE 100

int q[SIZE];

int f=-1,r=-1;

void enqueue(int x)

{ if(f==-1 && r==-1)

{ f++;

r++;

q[f]=x;

return;

}

if(r==SIZE-1)

{ printf("OVERFLOW\n");

return;}

r++;

q[r]=x;

}

void dequeue()

{ if(f==-1 && r==-1)

{ printf("UNDERFLOW\n");

return;}

if(f==r)

{ printf("THE DELETED ELE %d\n",q[f]);

f=r=-1;

return;}

printf("The deleted element is %d\n",q[f]);

f++;

}

void display()

{ for(int i=f;i<=r;i++)

{ printf("%d ",q[i]);

}

printf("\n");}

printf("1 TO ENQUEUE\n2 TO DEQUEUE\n3 TO DISPLAY\n");

do

{ printf("ENTER YOUR CHOICE ");

scanf("%d",&ch);

switch(ch)

{ case 1:

int x;

printf("ELEMENT TO BE ADDED");

scanf("%d",&x);

enqueue(x);

break;

case 2:

dequeue();

break;

case 3:

display();

break;

default:

break;

} } while(ch<=3);

printf("THANK YOU");

}