**CODE:-**

#include <stdio.h>

#include <stdlib.h>

typedef struct node node;

struct node

{

int info;

node \*link;

} \*front = NULL, \*rear = NULL;

void insert(int);

int del();

int peek();

void display();

int main()

{

int ch, num;

while (1)

{

printf("Enter 1 to insert.\n");

printf("Enter 2 to delete.\n");

printf("Enter 3 to peek at first element.\n");

printf("Enter 4 to display all elements of queue.\n");

printf("Enter 5 to exit.\n");

scanf("%d", &ch);

switch (ch)

{

case 1:

printf("Enter the number to be inserted.\n");

scanf("%d", &num);

insert(num);

break;

case 2:

printf("Enter 2 to delete an element.\n");

num = del();

printf("The deleted element was %d.\n", num);

break;

case 3:

num = peek();

printf("The topmost element is %d.\n", num);

break;

case 4:

display();

break;

case 5:

exit(1);

default:

printf("Erroneous input.\n");

}

}

}

int isEmpty()

{

if (front == NULL)

return 1;

else

return 0;

}

void insert(int item)

{

node \*tmp;

tmp = (node \*)malloc(sizeof(node));

if (tmp == NULL)

{

printf("Memory not available");

return;

}

tmp->info = item;

tmp->link = NULL;

if (front == NULL)

front = tmp;

else

rear->link = tmp;

rear = tmp;

}

int del()

{

node \*tmp;

int item;

if (isEmpty())

{

printf("Queue Underflow\n");

exit(1);

}

tmp = front;

item = tmp->info;

front = front->link;

free(tmp);

return item;

}

int peek()

{

if (isEmpty())

{

printf("Queue Underflow\n");

exit(1);

}

return front->info;

}

void display()

{

node \*tmp;

tmp = front;

if (isEmpty())

{

printf("Queue is empty\n");

return;

}

printf("The queue elements are :\n");

while (tmp != NULL)

{

printf("%d ", tmp->info);

tmp = tmp->link;

}

printf("\n");

}

**OUTPUT:-**

Enter 1 to insert.

Enter 2 to delete.

Enter 3 to peek at first element.

Enter 4 to display all elements of queue.

Enter 5 to exit.

1

Enter the number to be inserted.

5

Enter 1 to insert.

Enter 2 to delete.

Enter 3 to peek at first element.

Enter 4 to display all elements of queue.

Enter 5 to exit.

1

Enter the number to be inserted.

3

Enter 1 to insert.

Enter 2 to delete.

Enter 3 to peek at first element.

Enter 4 to display all elements of queue.

Enter 5 to exit.

1

Enter the number to be inserted.

7

Enter 1 to insert.

Enter 2 to delete.

Enter 3 to peek at first element.

Enter 4 to display all elements of queue.

Enter 5 to exit.

3

The topmost element is 5.

Enter 1 to insert.

Enter 2 to delete.

Enter 3 to peek at first element.

Enter 4 to display all elements of queue.

Enter 5 to exit.

4

The queue elements are :

5 3 7

Enter 1 to insert.

Enter 2 to delete.

Enter 3 to peek at first element.

Enter 4 to display all elements of queue.

Enter 5 to exit.

2

Enter 2 to delete an element.

The deleted element was 5.

Enter 1 to insert.

Enter 2 to delete.

Enter 3 to peek at first element.

Enter 4 to display all elements of queue.

Enter 5 to exit.

3

The topmost element is 3.

Enter 1 to insert.

Enter 2 to delete.

Enter 3 to peek at first element.

Enter 4 to display all elements of queue.

Enter 5 to exit.

4

The queue elements are :

3 7

Enter 1 to insert.

Enter 2 to delete.

Enter 3 to peek at first element.

Enter 4 to display all elements of queue.

Enter 5 to exit.

2

Enter 2 to delete an element.

The deleted element was 3.

Enter 1 to insert.

Enter 2 to delete.

Enter 3 to peek at first element.

Enter 4 to display all elements of queue.

Enter 5 to exit.

2

Enter 2 to delete an element.

The deleted element was 7.

Enter 1 to insert.

Enter 2 to delete.

Enter 3 to peek at first element.

Enter 4 to display all elements of queue.

Enter 5 to exit.

2

Enter 2 to delete an element.

Queue Underflow