CLASS XII
SUBJECT – COMPUTER SCIENCE
TOPIC - DATA STRUCTURES

Q1. IMPLEMENT STACK AS LINKED LIST AND STORE THE OBECT OF TYPE DISTANCE CLASS(METRES , CENTIMETRES) IN EACH NODE.
PROGRAM MUST HAVE PUSH AND POP FUNCTIONS THAT INSERT AND DELETE A NODE FROM THE STACK RESPECTIVELY.
THERE SHOULD BE A TOP POINTER THAT POINTS TO THE TOPMOST ELEMENTS OF THE STACK.
YOUR PROGRAM MUST CONSIDER ALL CASES THAT CAN LEAD TO AN ERROR.
PROGRAM SHOULD BE MENU DRIVEN THAT CONTINUES TILL THE USER WANTS.
DECLARE MEMBER FUNCTIONS IN THE CLASS DISTANCE AS PER YOUR REQUIREMENT.
PROGRAM SHOULD PRINT NEW STACK AFTER EACH OPERATION.

Q2. IMPLEMENT STACK AS ARRAY OF MAXIMUM DIMENSION 10 AND STORE THE OBECT OF TYPE DISTANCE CLASS(METRES , CENTIMETRES) AS EACH ELEMENT OF THE ARRAY..
PROGRAM MUST HAVE PUSH AND POP FUNCTIONS THAT INSERT AND DELETE AN ELEMENT FROM THE STACK RESPECTIVELY.
THERE SHOULD BE A TOP VARIABLE THAT STORES THE INDEX OF THE TOPMOST ELEMENT OF THE STACK.
YOUR PROGRAM MUST CONSIDER ALL CASES THAT CAN LEAD TO AN ERROR.
PROGRAM SHOULD BE MENU DRIVEN THAT CONTINUES TILL THE USER WANTS.
DECLARE MEMBER FUNCTIONS IN THE CLASS DISTANCE AS PER YOUR REQUIREMENT.
PROGRAM SHOULD PRINT NEW STACK AFTER EACH OPERATION

Q3 IMPLEMENT LINEAR QUEUE AS LINKED LIST AND STORE THE OBECT OF TYPE DISTANCE CLASS(METRES , CENTIMETRES) IN EACH NODE.
PROGRAM MUST HAVE INSERT AND DELETE FUNCTIONS THAT INSERT AND DELETE A NODE FROM THE QUEUE RESPECTIVELY.
THERE SHOULD BE A FRONT AND REAR POINTERS THAT POINTS TO THE FIRST AND LAST ELEMENTS OF THE QUEUE RESPECTIVELY.
YOUR PROGRAM MUST CONSIDER ALL CASES THAT CAN LEAD TO AN ERROR.
PROGRAM SHOULD BE MENU DRIVEN THAT CONTINUES TILL THE USER WANTS.
DECLARE MEMBER FUNCTIONS IN THE CLASS DISTANCE AS PER YOUR REQUIREMENT.
PROGRAM SHOULD PRINT NEW QUEUE AFTER EACH OPERATION

Q4 IMPLEMENT LINEAR QUEUE AS ARRAY OF MAXIMUM DIMENSION 10 AND STORE THE OBECT OF TYPE DISTANCE CLASS(METRES , CENTIMETRES) AS EACH ELEMENT OF THE ARRAY..
PROGRAM MUST HAVE INSERT AND DELETE FUNCTIONS THAT INSERT AND DELETE A NODE FROM THE QUEUE RESPECTIVELY.
THERE SHOULD BE A FRONT AND REAR VARIABLE THAT CONTAINS THE INDEX OF THE FIRST AND LAST ELEMENT OF THE QUEUE RESPECTIVELY.
YOUR PROGRAM MUST CONSIDER ALL CASES THAT CAN LEAD TO AN ERROR.
PROGRAM SHOULD BE MENU DRIVEN THAT CONTINUES TILL THE USER WANTS.
DECLARE MEMBER FUNCTIONS IN THE CLASS DISTANCE AS PER YOUR REQUIREMENT.
PROGRAM SHOULD PRINT NEW QUEUE AFTER EACH OPERATION.

Q5 IMPLEMENT CIRCULAR QUEUE AS LINKED LIST AND STORE THE OBECT OF TYPE DISTANCE CLASS(METRES , CENTIMETRES) IN EACH NODE.
PROGRAM MUST HAVE INSERT AND DELETE FUNCTIONS THAT INSERT AND DELETE A NODE FROM THE QUEUE RESPECTIVELY.
THERE SHOULD BE A FRONT AND REAR POINTERS THAT POINTS TO THE FIRST AND LAST ELEMENTS OF THE QUEUE RESPECTIVELY.
YOUR PROGRAM MUST CONSIDER ALL CASES THAT CAN LEAD TO AN ERROR.
PROGRAM SHOULD BE MENU DRIVEN THAT CONTINUES TILL THE USER WANTS.
DECLARE MEMBER FUNCTIONS IN THE CLASS DISTANCE AS PER YOUR REQUIREMENT.
PROGRAM SHOULD PRINT NEW QUEUE AFTER EACH OPERATION

Q6 IMPLEMENT CIRCULAR QUEUE AS ARRAY OF MAXIMUM DIMENSION 10 AND STORE THE OBECT OF TYPE DISTANCE CLASS(METRES , CENTIMETRES) AS EACH ELEMENT OF THE ARRAY..
PROGRAM MUST HAVE INSERT AND DELETE FUNCTIONS THAT INSERT AND DELETE A NODE FROM THE QUEUE RESPECTIVELY.
THERE SHOULD BE A FRONT AND REAR VARIABLE THAT CONTAINS THE INDEX OF THE FIRST AND LAST ELEMENT OF THE QUEUE RESPECTIVELY.
YOUR PROGRAM MUST CONSIDER ALL CASES THAT CAN LEAD TO AN ERROR.
PROGRAM SHOULD BE MENU DRIVEN THAT CONTINUES TILL THE USER WANTS.
DECLARE MEMBER FUNCTIONS IN THE CLASS DISTANCE AS PER YOUR REQUIREMENT.
PROGRAM SHOULD PRINT NEW QUEUE AFTER EACH OPERATION.

Q7
```
struct node {int x,y; };
class queue { node *front, *rear ;
public :
queue( ) { -----}
void insertq( ){-------}
void deleteq( ){-----}
~queue(){----}
};
```

COMPLETE THE CLASS DEFINITION PERTAINING TO LINKED QUEUE.

Q8
```
struct node {int x,y; };
class queue { node *front, *rear ;
public :
queue( ) { -----}
void insertq( ){-------}
void deleteq( ){-----}
~queue(){----}
};
```

COMPLETE THE CLASS DEFINITION PERTAINING TO CIRCULAR LINKED QUEUE.

Q9
```
struct node {int x,y; };
class queue { node front, rear ;
public :
queue( ) { -----}
void insertq( ){-------}
void deleteq( ){-----}
~queue(){----}
};
```

COMPLETE THE CLASS DEFINITION PERTAINING TO STATIC QUEUE.

Q10
```
struct node {int x,y; };
class queue { node front, rear ;
public :
queue( ) { -----}
void insertq( ){-------}
void deleteq( ){-----}
~queue(){----}
};
```

COMPLETE THE CLASS DEFINITION PERTAINING TO STATIC CIRCULAR QUEUE

Q11
```
struct node {int x,y; };
class stack{ node top ;
public :
stack( ) { -----}
void inserts( ){-------}
void deletes( ){-----}
~stack(){----}
};
```

COMPLETE THE CLASS DEFINITION PERTAINING TO STATIC STACK

Q12
```
struct node {int x,y; };
class stack{ node *top ;
public :
stack( ) { -----}
void inserts( ){-------}
void deletes( ){-----}
~stack(){----}
};
```

COMPLETE THE CLASS DEFINITION PERTAINING TO LINKED STACK