**PSG COLLEGE OF TECHNOLOGY, COIMBATORE-04**

**DEPARTMENT OF APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCES**

**II Semester MSc Software Systems**

**18XW26 Data Structures Lab**

**STACKS, QUEUE & CIRCULAR QUEUE**

**Problem Sheet – 5**

1. Define a class STACK with top (integer variable) and stack (an integer array) as data members. Define a default constructor to initialize top by -1. Write the following member functions
2. Push() - insert an element into the stack
3. Pop () - delete an element from the stack
4. Display\_from\_bottom() – print the stack from the bottom of the stack

To print the stack from the bottom, we must pop each element and push it onto another stack. The characters can then be popped from the second stack and printed in the order they are popped.

1. Display\_from\_top() – print the stack from the top of the stack
2. Count() – print the number of elements currently in the stack

Write a C++ menu driven program to list the above task and let the user to select one from the list. The program should be terminated when the user explicitly select the exit option.

1. A text editor can processes a line of text using stack. The editor reads one character at a time. If the entered character is the erase character(‘#’), the editor pops the stack, otherwise it pushes the character onto the stack. Write a C++ code to execute these actions.

Sample input/output:

ABC#RY#WA###

The elements in the stack are: AB

1. It is possible to keep two stacks in a single array, if one grows from end position of the array, and the other grows from the opposite direction.

Define a class with following member functions,

1. PUSH(x,s) – pushes element x onto stack s, where s is one or the other of these two stacks. ( s can be 1 or 2, 1 indicates the insertion to be in 1st stack (right direction), 2 denotes 2nd stack (growing in left direction)).
2. POP(s) – pops an element from the stack s
3. HIGH() – displays the elements of the stack which has highest number of elements
4. LOW() – displays the elements of the stack which has lowest number of elements
5. An organization XYZ has the car parking area as only one car can run through it at a time. If three cars are standing in the parking area, 4th car can stand after 3rd one. Suppose the owner of the second car wants to take his car, then he has to move (pop) 4th and 3rd cars from the car parking place and park them in outside(another stack). After getting his car, he has to move (push) again the 3rd and 4th cars in the parking area in the same order. Now 1st,3rd and 4th cars are standing in the car parking place.

Write a class with following member functions

1. Push (car number) – pushes the car number of a car into the stack
2. Pop(car number) – pops the specified car from the parking area
3. Count()- displays the number of cars currently parked in the parking area

Note: when you want to pop intermediate element of a stack, you should pop the elements from top till to reach the element which you want to delete and push the popped elements into another stack. Once the element has been deleted then pop the elements from the second stack and push them into the first one to get the same order of the elements as previous except the deleted one.

1. Define a class QUEUE with front (integer variable), rear (integer variable) and queue (an integer array) as data members. Define a default constructor to initialize rear and front by -1. Write the following member functions
2. Enqueue() - insert an element into the queue
3. Dequeue() - delete an element from the queue
4. Display() – print the queue from front to rear of the queue

To print the queue from front to rear, we should delete each element from front, to display that element. The deleted elements can be maintained in another queue, for further reference.

1. Count() – print the number of elements currently in the queue

Write a C++ menu driven program to list the above task and let the user to select one from the list. The program should be terminated when the user explicitly select the exit option.

1. Define a class CIR\_QUEUE with front (integer variable), rear (integer variable) and queue (an integer array) as data members. Define a default constructor to initialize rear and front by 0. Write the following member functions
2. Insert() - insert an element into the circular queue
3. Delete () - delete an element from the circular queue
4. Display() – print the queue from front to rear of the queue

To print the queue from front to rear, we should delete each element from front, to display that element. The deleted elements can be maintained in another queue, for further reference.

1. Count() – print the number of elements currently in the circular queue

Write a C++ menu driven program to list the above task and let the user to select one from the list. The program should be terminated when the user explicitly select the exit option.