Lab -03

Date 1 / 1 / 24
Page

2. WAP to Simulate the working of queue of integers using an array. Provide following Operations.
ⓐ Insert ⓑ Delete ⓒ Display
The program should print appropriate messages for queue empty & queue overflow conditions.

AB+CD*-

```
→ ## include <stdio.h>
# include <Conio.h>
# define Max 10 // changing this value will change
                   length of array.
int queue [MAX];
int front = -1, rear = -1;
void insert (void);
int delete_element (void);
int peek (void);
void display (void);
int main() {
    int option, val;
    do {
        printf("\n\n ***MAIN MENU***");
        printf("\n 1. Insert an element");
        printf("\n 2. Delete an element");
        printf("\n 3. Peek");
        printf("\n 4. Display the queue");
        printf("\n 5. EXIT");
        printf("\n Enter your option: ");
        Scanf("%d", &option);
        Switch (option) {
        case1:
            insert();
            break
Case 2;
    val = delete_element();
    if (val != -1)
    printf("\n The number deleted is:
            %d", val);
    break;
  Case 3 :
     val = peek();
     if (val != -1)
     printf("\n The first value in queue
            is: %d", val);
     break;
  case 4 :
       display();
       break;
  }
}while(option != 5);
getch();
return 0;
}
void insert () {
     int num;
     printf ("\n Enter the number to be inserted
     in the queue: ");
     scanf("%d", &num);
     if (rear == MAX-1)
      printf("\n Overflow");
      else if(front == -1 && rear == -1)
      front = rear = 0;
      else
      rear ++;
      queue[rear] = num.
```

```
int i=0, j=0;
char temp;
strcpy(target, " ");
while (source[i]! ='\0') {
        if (source[i] == '(') {
              push(st, source[i]);
              i++;
        }
        else if (source[i] == ')') {
            while((top! = -1) && (st[top] ! ='(')){
                 target[j] = pop(st);
                 j++;
            }
            if (top == -1) {
                 printf("\n Incorrect Expression");
                 exit(1);
            }
          temp = pop(st);
          i++;
      }
      else if (isdigit(source[i]) || isalpha(sour
                                              -ce[i])){
          target[j] = Source[i];
          j++;
          i++;
      }
      else if (source[i] == '+'|| source[i]=='-'
      || source[i] =='*' || source [i] == '/' ||
               source[i] == '%') {
            while((top != -1) && (st[top] != '(')
&& (getPriority (st[top]) > getPriority (source[i])){
               Target[j] = pop(st);
               j++;
            push(st, source[i]);
            i++;
        }
        else {
            printf("\n Incorrect element In Expression
                                              ");
            exit(1);
        }
    }
    while((top!=-1) && (st[top]!='(')) {
        target[j] = pop(st);
        j++;
    }
    target[j] = '\0';
}
int getPriority(char op){
    if(op=='/' || op == '*' || op == '%')
        return 1;
    else if(op=='+' || op=='-')
        return 0;
} }
void push(char st[], char val) {
    if(top == MAX-1)
        printf("\n Stack Overflow");
    else {
        top++;
        st[top] = val;
    }
}
char pop(char st[]) {
    char val=' ';
    if(top==-1)
        printf("\n Stack underflow");
```

3 WAP to Simulate the working of a circular queue of integers using an array. Provide the following Operations.
ⓐ Insert ⓑ delete ⓒ Display
The program should print appropriate messages for queue empty & queue overflow conditions

```
→ # include <stdio.h>
  # include < Conio.h>
  # define MAX 10
  int queue [MAX];
  int front = -1, rear = -1;
  void insert (void);
  int delete_element (void);
  int peak (void);
  void display (void);

  int main () {
     int option, val;
     clrscr ()
     do {
        printf ("\n *** MAIN MENU*** );
        printf ("\n 1. Insert an element ");
        printf ("\n 2. Delete an element ");
        printf ("\n 3. peek");
        printf ("\n 4. Display the queue");
        printf ("\n 5. Exit ");
        printf ("\n Enter your option: ");
        scanf ("%d", &option);
        switch (option) {
              case 1 :
                    insert ();
                    break;
```

```
            Case 2:
                val = delete_element ();
                if (val! = -1)
                    printf ("\n The no%. deleted is : %d",
                              val);
               break;
         Case 3:
            val = peek();
            if (val ! = -1)
               printf ("\n The first value in queue
                    is: %d", val);
            break;
       Case 4:
            display ();
            break;
   }
 } while (option != 5);
   getch ();
   return 0;
}
void insert () {
     int num;
     printf ("\n Enter no%. to be inserted in the
            queue: ");
    Scanf ("%d", &num);
    if (front ==0 && rear == MAX-1)
           printf ("\n Overflow");
    else if (front == -1 && rear == -1) {
           front = rear = 0;
           queue [rear] = num;
   }
```

```
  else if(rear == MAX-1 && front != 0){
      rear = 0;
      queue [rear] = num;
  }
  else {
      rear ++;
      queue [rear] = num;
  }
}
int delete_element() {
      int val;
      if (front == -1 && rear == -1) {
          printf("\n underflow");
          return -1;
      }
      else val = queue[front];
      if(front == rear)
              front = rear = -1;
      else {
          if(front == MAX-1)
                  front = 0;
          else
               front ++;
      }
      return val;
}
int peek(){
   if (front == -1 && rear == -1){
      printf("\n Queue is Empty");
      return -1;
   }
   else {
      return queue [front];
   }
```

```
void display() {
        int i;
        printf("\n");
        if(front == -1 && rear == -1)
              printf("\n Queue is Empty");
        else {
            if(front < rear) {
                for(i = front ; i <= rear ; i++)
                      print(" \t %d", queue[i]);
            }
            else {
               for(i = front ; i < MAX ; i++)
                      print("\t %d", queue[i]);
               for(i = 0 ; i <= rear ; i++)
                      print("\t %d", queue[i]);
            }
         }
}
```

1/1/24

Output

```
 *** MAIN MENU ***
 1. Insert an element
 2. Delete an element
 3. Peek
 4. Display the queue
 5. Exit
 Enter your option : 1
 Enter no. to be inserted in queue: 35
  Enter you option : 2
Enter no. to be delete : 25
 Enter your option : 3
 Queue is Empty
```