Date 28/12/23
Page

Lab 2:

1) Infix to Postfix conversion

```
# include <stdio.h>
# include <ctype.h>
# define SIZE 50

char stack [SIZE];
int top = -1;

push (char elem)
{
    Stack [++top = elem;
}
char pop ()
{
    return (stack [top--]);
}
int pr (char symbol)
{
    if (symbol == '^')
    {
        return (3);
    }
    else if (symbol == '*' || symbol == '/')
    {
        return (2);
    }
    else if (symbol == '+' || symbol == '-')
    {
        return (1);
    }
    else {
```

```
    return (0);
  }
}
void main()
{
  char infix[50], postfix[50], ch, elem;
  int i=0, k=0;
  printf("Enter the Infix expression:");
  scanf("%s", infix);
  push("#");
  while ((ch=infix[i++])!='\0')
  {
    if (ch=='(') push(ch);
    else
      if (isalnum(ch)) postfix[k++]=ch;
    else
      if (ch==')')
    {
      while (stack[top]!='(')
        postfix[k++]=pop();
      postfix[k]='\0';
      printf("\n postfix expression: %s\n", postfix);
}.  }.
```

Output:-

Enter the infix expression : A*B+C*D-E

Postfix expression : AB*CD*+E-

Date __/__/__
Page _____

Pgm 2:- Postfix Evaluation Code

```
#include <stdio.h>
int stack[20];
int top = -1;

void push (int x)
{
    stack[++top] = x;
}
int pop ()
{
    return stack[--top];
}
int main ()
{
    char exp[20];
    char *e;
    int a,b,c,num;
    printf ("Enter the expressions :: ");
    scanf ("%s", exp);
    e = exp;
    while (*e != '\0')
    {
        if (isdigit (*e))
        {
            num = *e - 48;
            push (num);
        }
        else {
            a = pop();
            b = pop();
            switch (*e)
```

```
        {
            case '+':
            {
                c=a+b;
                break;
            }
            case '-':
            {
                c=a-b;
                break;
            }
            case '*':
            {
                c=a*b;
                break;
            }
            case '/':
            {
                c=a/b;
                break;
            }
        }
        push(c);
    }
    e++;
}
printf("The result of expression is %s = %d",
                    exp, pop());
}
```

Date __/__/__
Page _____

Output :-

Enter the expression : 12* 34* +5 -

The result of expression 12* 34* +5 = 9

Q.1.

28/12/23

Lab 3 programs:

Queue:-

```
#include <stdio.h>
#define MAX 50
int queue_array[MAX];
int rear = -1;
int front = -1;
display ()
{
   int i;
   if (front == -1)
      printf ("Queue is empty \n");
   else
   {
      printf ("Queue is: \n");
      for (i = front; i <= rear; i++)
         printf ("%d", queue_array[i]);
      printf ("\n");
   }
}
void main ()
{
   int choice;
   while (1)
{
   printf ("1. Insert \n");
   printf ("2. Delete \n");
   printf ("3. Display \n");
   printf ("4. Exit \n");
   printf ("Enter your choice \n");
   scanf ("%d", &ch);
   switch (ch)
   {
```

Date __/__/__
Page _____

```
case 1: insert ();
        break;
case 2: delete ();
        break;
case 3: display ();
        break;
case 4: exit (1);
        break;
default :
    printf ("Invalid \n");
}
}
}
void insert (int x)
{
  if (rear == MAX-1)
  {
    printf ("Overflow");
  }
  else if (rear == -1 && front == -1)
  {
    front = rear = 0;
    Printf ("Insert element to queue :");
    queue[rear] = x;        scanf("%d", &x);
  }
  else
  {
    rear ++;
    queue[rear] = x;
  }
}
```

```
void delete().
{
    if (front == -1 || rear == -1).
    {
        printf ("Underflow");
    }.
else if (front == rear).
    front = rear = -1;
else.
    {
        printf ("Element to be deleted: %d", queue_array
                                                [front]);
        front++;
    }.
}.
```

Circular queue:-

```
# include <stdio.h>.
# define MAX 50.
int queue_array[MAX];
int front = -1;
int rear = -1;
void enqueue();
void dequeue();
void display();

void main().
{
    int ch;
    while (1).
    {
```

Date __/__/__
Page _____

```
printf ("1. Enqueue \t");
printf ("2. Dequeue \t");
printf ("3. Display \t");
printf ("4. Exit \t");
printf ("Enter your choice \n");
scanf ("%d", &ch);
switch (ch)
{
    case 1: enqueue ();
            break;
    case 2: dequeue ();
            break;
    case 3: display ();
            break;
    case 4: exit (1);
            break;
    default: printf ("Invalid");
}
}
void enqueue (int x)
{
    if (front == -1 && rear == -1)
        printf ("Queue is full \n");
    else
    {
        if (front == -1)
        front = 0;
        rear = (rear + 1) % MAX;
        queue [rear] = x;
        printf ("inserted %d", x);
    }
}
```

```
void dequeue ()
{
    int x;
    if (isEmpty ()) (front == -1 && rear == -1).
        printf ("Empty ");
    else if (front == rear)
        {
            front = rear = -1;
        }.
    else {
            printf ("%d", "Element to be deleted %d",
                    queue (front));
            front = (front + 1) %. MAX;
        }.
}.
void display ()-
{
    int i = front;
    if (front == -1 && rear == -1).
            printf ("Underflow");
    else.
    {
        printf ("Queue is :");
        while (i != rear).
        {
            printf ("%d", queue [i]);
            i = (i + 1) %. MAX;
        }.
        printf ("%d", queue [rear]);
    }
}.
```

Date __/__/__
Page _____

OUTPUT:-

1. Enqueue 2. Dequeue 3. Display 4- Exit.

Enter your choice :
1.
Enter the number to be inserted into the queue
4

1. Enqueue 2. Dequeue 3. Display 4-Exit

Enter your choice : 2.
4 was removed from queue.

1. Enqueue 2. Dequeue. 3. Display 4-Exit.
Enter your choice :3
Queue is empty.