28/12/23 Week - 3

Lab program 2

Write a program to convert a given valid parenthesized infix arithmetic expression to postfix expression. The expression consists of single character operands and the binary operators + (plus), - (minus), * (multiply), / (divide) and ^ (power).

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#define MAX 100

char stack[MAX];
char infix[MAX];
char postfix[MAX];
int top=-1;

void push(char);
char pop();
int isEmpty();
void InToPost();
void print();
int precedence(char);

int main()
{
   printf("Enter infix expression: ");
   gets(infix);
   InToPost();
   print();
```

Evaluated 18/1/24

```
return 0;
}
void InToPost()
{
  int i, j=0;
  char symbol, next;
  for(i=0; i<strlen(infix); i++)
  {
    symbol = infix[i];
    switch(symbol)
    {
      case '(':
        push(symbol);
        break;
      case ')':
        while((next=pop())!='(')
          postfix[j++]=next;
          break;
      case '+':
      case '-':
      case '*':
      case '/':
      case '^':
        while(!isEmpty() && precedence(stack
          [top])>=precedence(symbol))
            postfix[j++]=pop();
        push(symbol);
        break;
      default:
        postfix[j++]= symbol;
    }
}
```

```
while (!isEmpty())
        postfix[j++] = pop();
postfix[j] = '\0';
}

int precedence(char symbol)
{
   switch(symbol)
    {
     case '^':
          return 3;
     case '/':
     case '*':
          return 2;
    case '+':
    case '-':
          return 1;
    default:
          return 0;
    }
}

void print()
{
 int i=0;
 printf("The equivalent postfix expression is:");
  while(postfix[i])
   {
     printf("%c", postfix[i++]);
   }
   printf("\n");
}
```

PAGE EDGE
DATE : / /

```
void push(char c)
{
  if(top==MAX-1)
  {
    printf("stack overflow");
    return;
  }
  top++;
  stack[top]=c;
}
char pop()
{
  char c;
  if(top==-1)
  {
    printf("stack overflow");
    exit(1);
  }
  c=stack[top];
  top=top-1;
  return c;
}
int isempty()
{
  if(top==-1)
    return 1;
  else
    return 0;
}
```

Output:

Enter Infix expression: ((a+b)-(c*d))
The equivalent postfix expression is:
ab+cd*-

2) Postfix evaluation:

3a)
```
#include<stdio.h>
#define MAX 50
void insert();
void delete();
void display();
int queue_array[max];
int rear=-1;
int front=-1;
main()
{
  int choice;
  while(1)
  {
  printf("1. insert element\n");
  printf("2. delete element\n");
  printf("3. display elements:\n");
  printf("4. quit\n");
  printf("Enter your choice:");
  scanf("%d", &choice);
  switch(choice)
  {
```

```
case 1:
  Insert();
  break;
case 2:
  delete();
  break;
case 3:
  display();
  break;
case 4:
  exit(1);
  default
    printf("wrong choice \n");
  }
}
}
void Insert()
{
  int add_item;
  if (rear==MAX-1)
  printf("Queue overflow \n");
.. else {
    if (front == -1)
    front=0;
    printf("Insert the element:");
    scanf("%d", &add_item);
    rear=rear+1;
    queue_array[rear]=add_item;
  }
}
void delete()
{
```

18/1/24

```
if (front==-1 || front>rear)
{
  printf("queue underflow \n");
    return;
  }
else
  {
  printf("element deleted from queue is:
          %d \n", queue_array[front]);
  front=front+1;
  }
}
void 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");
  }
}
```

Output

1. Insert element
2. Delete element
3. Display element
4. Quit.

DATE : / /

- Enter your choice : 1
  Insert element : 23
- Enter your choice : 1
  Insert element : 24
- Enter your choice : 3
  queue is 23, 24.
- Enter your choice : 2
  element deleted from queue is : 23
- Enter your choice : 4.