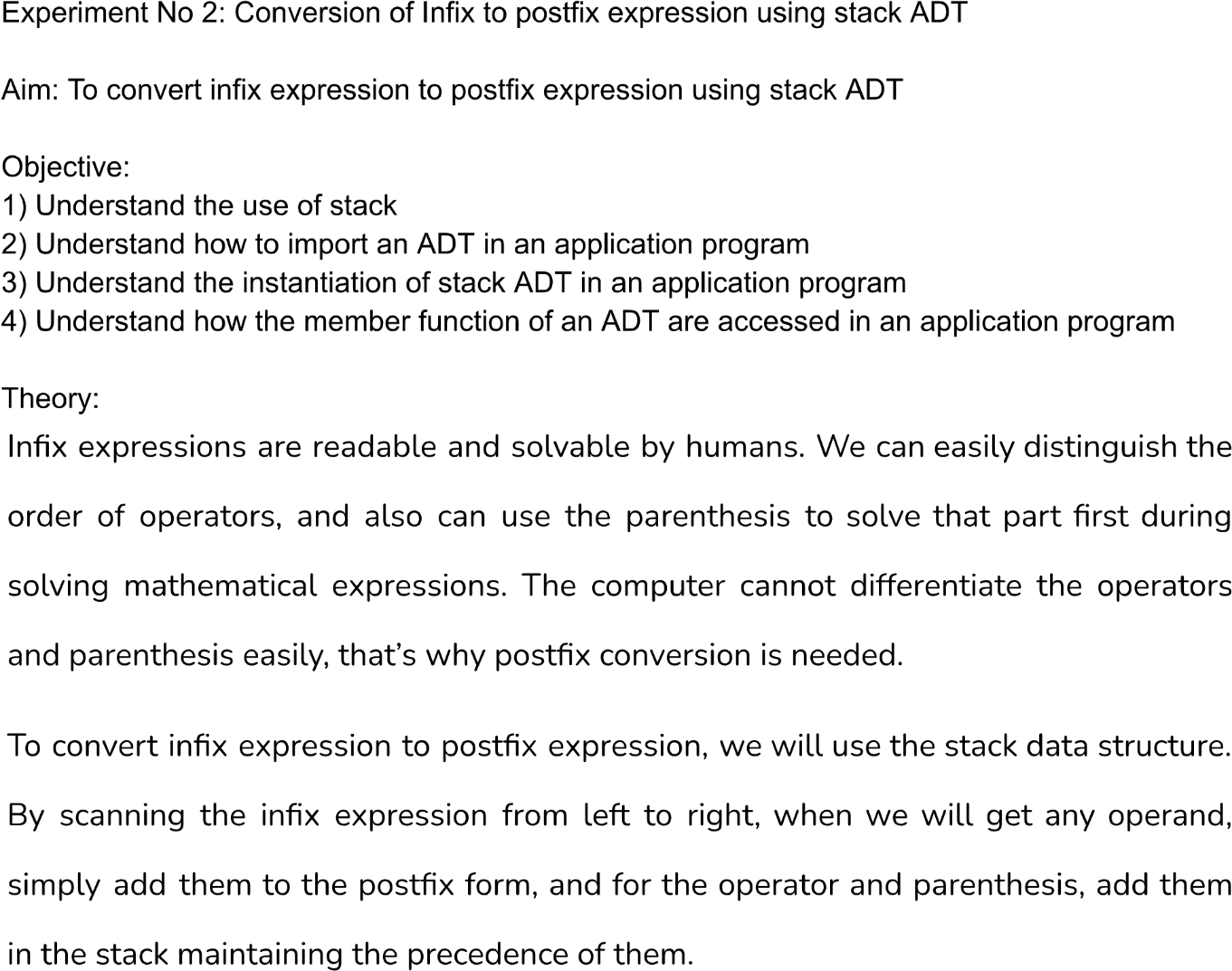
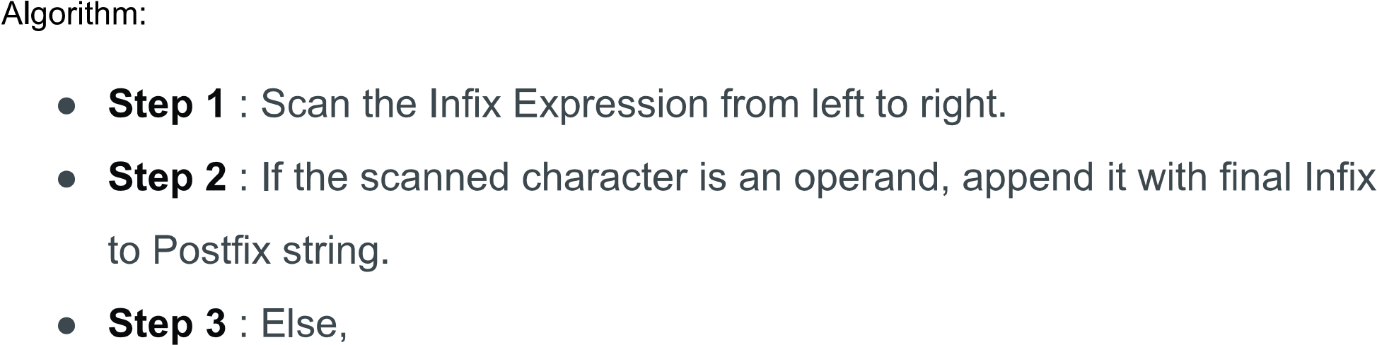
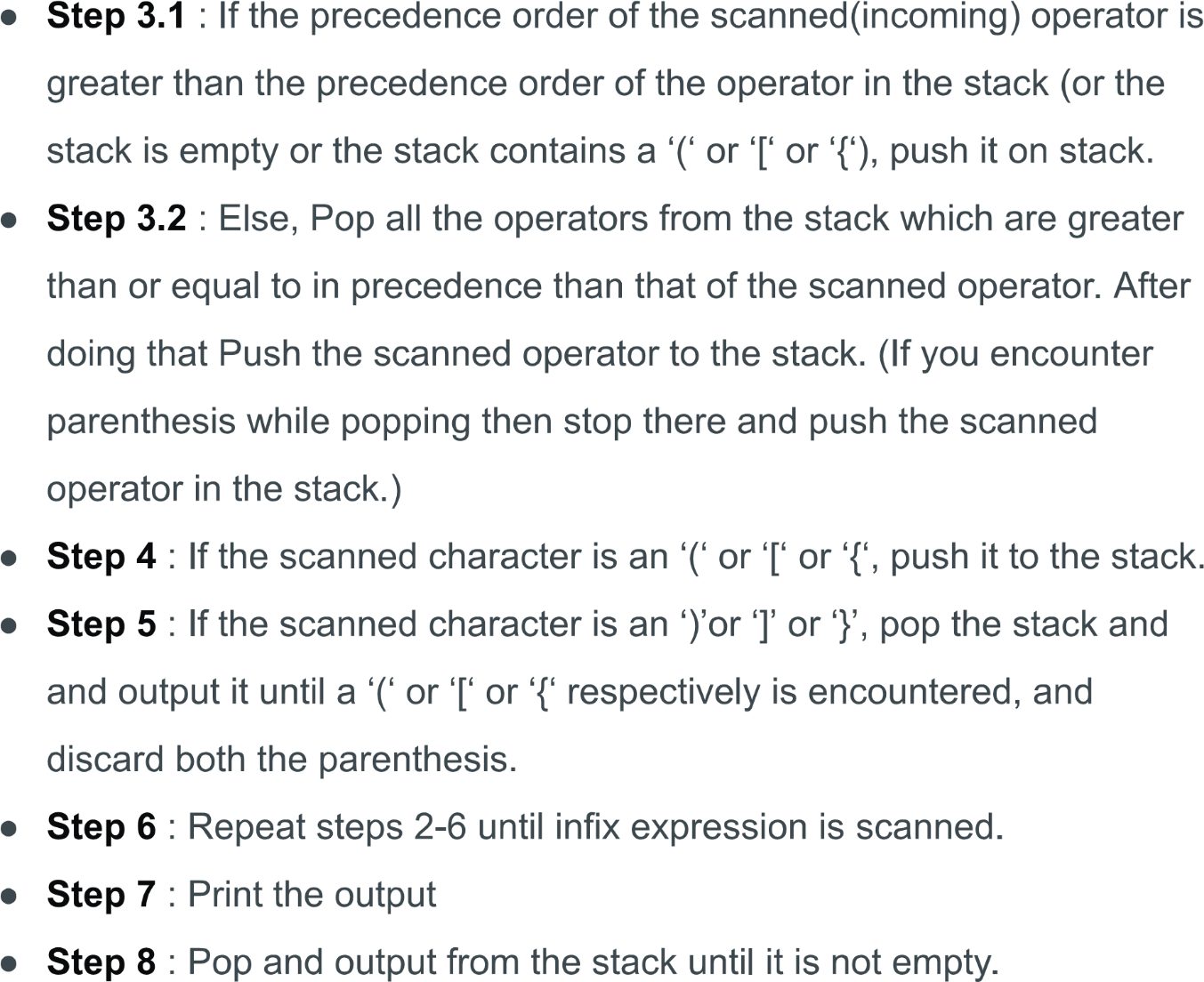
Name:Kaushal Pravin Tare Div:C Roll No:50







**Code:**

#include<stdio.h>

#include<ctype.h>

#include<string.h>

#include<stdlib.h>

#define size 100

char stack[size];

int top=-1;

void push(char item)

{

if(top>=size-1)

{

printf("\n stack verification");

}

else

{

top=top+1;

stack[top]=item;

}

}

char pop()

{

char item;

if(top<0)

{

printf("stack is underflow");

getchar();

exit(1);

}

else

{

item=stack[top];

top--;

return(item);

}

return 0;

}

int is\_operator(char symbol)

{

if(symbol=='^' || symbol=='+'|| symbol=='/'|| symbol=='\*' || symbol=='-')

{

return 1;

}

else

{

return 0;

}

}

int precedence(char symbol)

{

if(symbol=='^')

{

return (3);

}

else if(symbol=='\*'||symbol=='/')

{

return(2);

}

else if(symbol=='+'||symbol=='-')

{

return(1);

}

else

{

return(0);

}

}

void InfixToPostfix(char infix\_exp[],char postfix\_exp[])

{

int i,j;

char item;

char x;

push ('(');

strcat(infix\_exp,")");

i=0;

j=0;

item=infix\_exp[i];

while(item!='\0')

{

if(item=='(')

{

push(item);

}

else if(isdigit (item) || isalpha (item))

{

postfix\_exp[j]=item;

j++;

}

else if(is\_operator(item)==1)

{

x=pop();

while(is\_operator(x)==1 && precedence(x)>=precedence(item))

{

postfix\_exp[j]=x;

j++;

x=pop();

}

push(x);

push(item);

}

else if(item==')')

{

x=pop();

while(x!='(')

{

postfix\_exp[j]=x;

j++;

x=pop();

}

}

else

{

printf("\n invalid infix expression\n");

getchar();

exit(1);

}

i++;

item=infix\_exp[i];

}

if(top>0)

{

printf("\n invalid infix expression\n");

getchar();

exit(1);

}

postfix\_exp[j]='\0';

}

int main()

{

char infix[size],postfix[size];

printf("assumption the infix expression contain single letter variable and single digit constact omly\n");

printf("\n enter infix expression:");

gets(infix);

InfixToPostfix(infix,postfix);

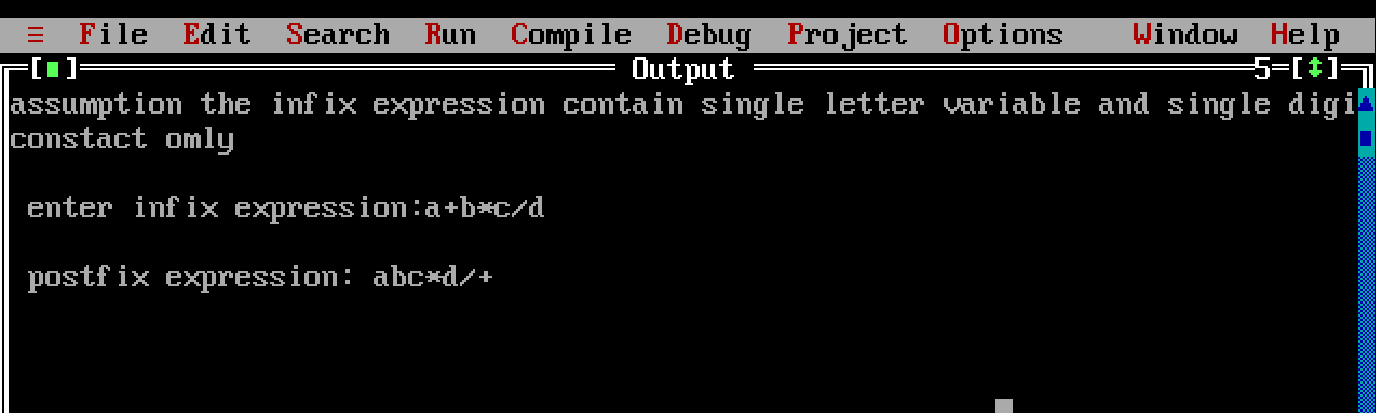
printf("\n postfix expression: ");

puts(postfix);

return 0;

}

Output:



Conclusion:

* Infix notation is the notation in which operators comes between the required operands.
* Postfix notation is the type of notation in which operators comes after the operand.
* Infix expression can be converted to postfix expression using stack.
* We saw how to convert infix to postfix by writing code in the language of c.