PRACTICE PROGRAMS

1 INFIX-PREFIX

#include<stdio.h>

#include<string.h>

#include<conio.h>

int F(char symbol)

{

switch(symbol)

{

case '+':

case '-':return(2);

case '\*':

case '/':return(4);

case '^':

case '$':return(5);

case '(':return(0);

case '#':return(-1);

default:return(8);

}

}

int G(char symbol)

{

switch(symbol)

{

case '+':

case '-':return(1);

case '\*':

case '/':return(3);

case '^':

case '$':return(6);

case '(':return(9);

case ')':return(0);

default:return(7);

}

}

void infix\_prefix(char infix[],char prefix[])

{

int top,i,j;

char s[30],symbol;

top=-1;

j=0;

s[++top]='#';

for(i=0;i<strlen(infix);i++)

{

symbol=infix[i];

while(F(s[top])>G(symbol))

{

prefix[j]=s[top--];

j++;

}

if(F(s[top])!=G(symbol))

s[++top]=symbol;

else

top--;

}

while(s[top]!='#')

prefix[j++]=s[top--];

prefix[j]='\0';

strrev(prefix);

}

void main()

{

char infix[20];

char prefix[20];

printf("enter infix expression \n");

gets(infix);

strrev(infix);

infix\_prefix(infix,prefix);

printf("prefix expression :\n");

puts(prefix);

getch();

}

2.EVALUATION OF POSTFIX EXPRESSION

#include<stdio.h>

#include<string.h>

#include<conio.h>

#include<math.h>

#include<ctype.h>

double compute(char sym,double op1,double op2)

{

switch(sym)

{

case '+':return(op1+op2);

case '-':return(op1-op2);

case '\*':return(op1\*op2);

case '/':return(op1/op2);

case '^':

case '$':return(pow(op1,op2));

}

}

void main()

{

double s[20],res,op1,op2;

int top=-1;

char postfix[20],symbol;

printf("enter postfix expression \n");

gets(postfix);

for(int i=0;i<strlen(postfix);i++)

{

symbol=postfix[i];

if(isdigit(symbol))a

s[++top]=symbol-'0';

else

{

op2=s[top--];

op1=s[top--];

res=compute(symbol,op1,op2);

s[++top]=res;

}

}

res=s[top--];

printf(" result : %f ",res);

getch();

}

3.RECURSION – FACTORIAL

#include<stdio.h>

#include<conio.h>

int fact(int n)

{

if(n==0)

return(1);

return(n\*fact(n-1));

}

void main()

{

int n;

printf("enter a number \n");

scanf("%d",&n);

int f=fact(n);

printf("factorial :%d ",f);

}

4.RECURSION-GCD

#include<stdio.h>

int hcf(int n1,int n2)

{

if(n2!=0)

return hcf(n2,n1%n2);

else

return n1;

}

void main()

{

int n1,n2;

printf("enter two numbers \n");

scanf("%d %d ",&n1,&n2);

int res=hcf(n1,n2);

printf("GCD : %d ",res);

}