//INTEGRATION

CODE:

#include<stdio.h>

//integration

int main()

{

//inputs

int n,one,two,three;

printf("enter the no.of terms in your equation");

scanf("%d",&n);

int po[n];

int px[n];

printf("input the co-efficents of your terms in the polynomial");

for(int i=0;i<n;i++)

{

//coefficent input

scanf("%d",&px[i]);

}

printf("input the powers of your respective co-efficent of the term in the polynomial");

for(int i=0;i<n;i++)

{

//power input

scanf("%d",&po[i]);

}

//calculation

for(int i=0;i<n;i++)

{

if(po[i]==-1)

{

one=px[i];

two=1;

printf("+(%d)ln(+(%d)x^%d) ",one,one,two);

}

else

{

one=px[i]/po[i];

two=po[i]+1;

printf("+(+(%d)/(+%d))x^%d ",px[i],two,two);

}

}

}

SAMPLE INPUT OUTPUR FOR INTEGRARION:

Text

Description automatically generated

//DERIVATIVES

SAMPLE INPUT OUTPUT FOR PRODUCT RULE:

CODE:

#include<stdio.h>

#include<string.h>

//product rule and quotient rule

int main()

{

char eq[1000];

int n1,n2,t;

printf("enter the no.of terms in your first polynomial\n");

scanf("%d",&n1);

printf("enter the no.of terms in your second polynomial\n");

scanf("%d",&n2);

printf("\nenter the equation");

scanf("%s",eq);

int l=strlen(eq);

char eq1[5],eq2[5];

for(int i=0;i<l;i++)

{

if(eq[i]=='\*')

{ int k=i+1;

for(int j=0;j<i;j++)

{

eq1[j]=eq[j];

}

int p=0;

for(int j=i+1;j<=l;j++)

{

eq2[p]=eq[k];

k++;

p++;

}

//EQ1

int m=0,n=3,d=0;

int pow[n];

int coef[n];

int l1=strlen(eq1);

//extraction for coeff

for(int i=0;i<n1;i++)

{

coef[m]=(int)(eq1[d]-'0');

m++;

d=d+5;

}

//extraction for power

for(int i=0;i<n1;i++)

{

pow[i]=(int)(eq1[n]-'0');

n=n+5;

}

//calculation for coeff

for(int i=0;i<n1;i++)

{

coef[i]=coef[i]\*pow[i];

}

//calculation for power

for(int i=0;i<n1;i++)

{

pow[i]=pow[i]-1;

}

//EQ2

m=0,n=3;

int pow1[n];

int coef1[n];

//extraction for coeff

for(int i=0;i<n2;i++)

{

if(i%5==0)

{

coef1[m]=(int)(eq2[i]-'0');

m++;

}

}

//extraction for power

for(int i=0;i<n2;i++)

{

pow1[i]=(int)(eq2[n]-'0');

n=n+5;

}

//calculation for coeff

for(int i=0;i<n2;i++)

{

coef1[i]=coef1[i]\*pow1[i];

}

//calculation for power

for(int i=0;i<n2;i++)

{

pow1[i]=pow1[i]-1;

}

//PRINTING

printf("(%s)",eq1);

for(int i=0;i<n1;i++)

{

printf("(+(%d)x^(%d))",coef[i],pow[i]);

}

printf(" + (%s)",eq2);

for(int i=0;i<n2;i++)

{

printf("(+(%d)x^(%d))",coef1[i],pow1[i]);

}

}

//QUOTIENT RULE

else if(eq[i]=='/')

{

int k=i+1;

for(int j=0;j<i;j++)

{

eq1[j]=eq[j];

}

int p=0;

for(int j=i+1;j<=l;j++)

{

eq2[p]=eq[k];

k++;

p++;

}

//EQ1

int m=0,n=3,d=0;

int pow[n];

int coef[n];

//extraction for coeff

for(int i=0;i<n1;i++)

{

coef[m]=(int)(eq1[d]-'0');

m++;

d=d+5;

}

//extraction for power

for(int i=0;i<n1;i++)

{

pow[i]=(int)(eq1[n]-'0');

n=n+5;

}

//calculation for coeff

for(int i=0;i<n1;i++)

{

coef[i]=coef[i]\*pow[i];

}

//calculation for power

for(int i=0;i<n1 ;i++)

{

pow[i]=pow[i]-1;

}

//EQ2

m=0,n=3;

int pow1[n];

int coef1[n];

//extraction for coeff

for(int i=0;i<n2;i++)

{

if(i%5==0)

{

coef1[m]=(int)(eq2[i]-'0');

m++;

}

}

//extraction for power

for(int i=0;i<n2;i++)

{

pow1[i]=(int)(eq2[n]-'0');

n=n+5;

}

//calculation for coeff

for(int i=0;i<n2;i++)

{

coef1[i]=coef1[i]\*pow1[i];

}

//calculation for power

for(int i=0;i<n2;i++)

{

pow1[i]=pow1[i]-1;

}

//PRINTING

printf("(%s)",eq2);

for(int i=0;i<n1;i++)

{

printf("(+(%d)x^(%d))",coef[i],pow[i]);

}

printf("-");

printf("(%s)",eq1);

for(int i=0;i<n2;i++)

{

printf("(+(%d)x^(%d))",coef1[i],pow1[i]);

}

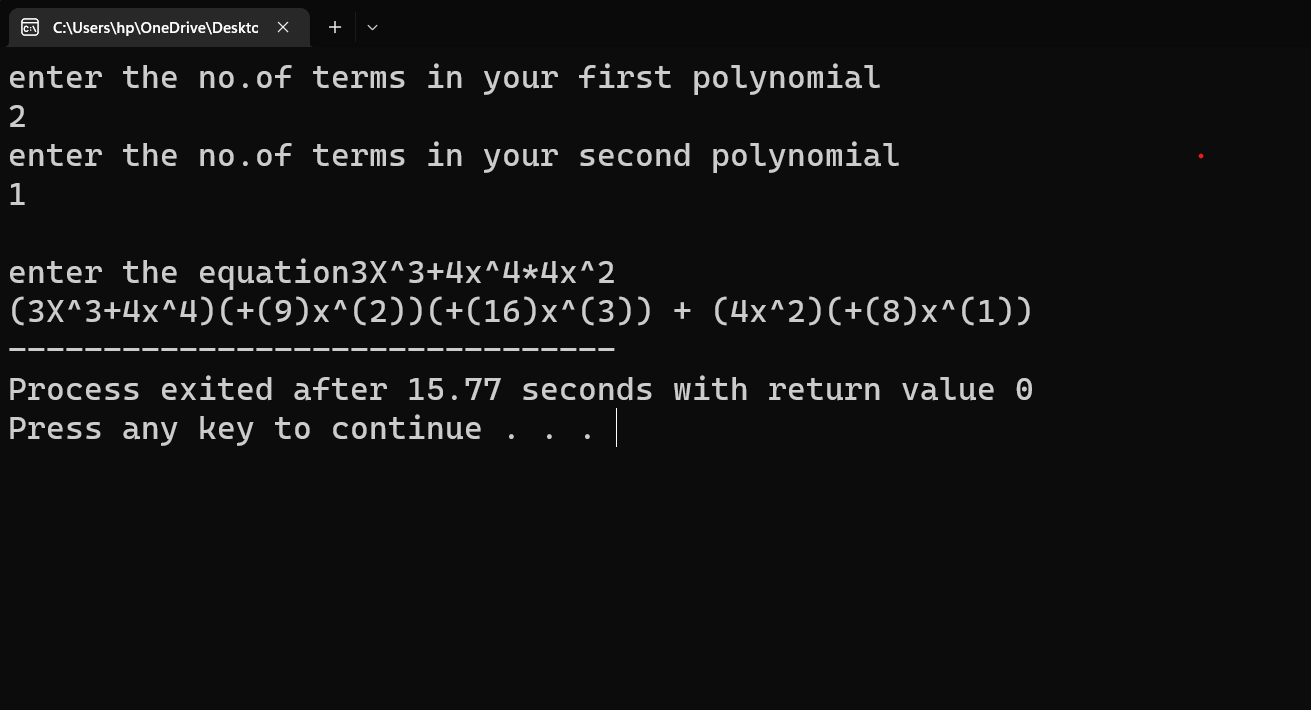
printf(" / ");

printf("(%s)^2",eq2);

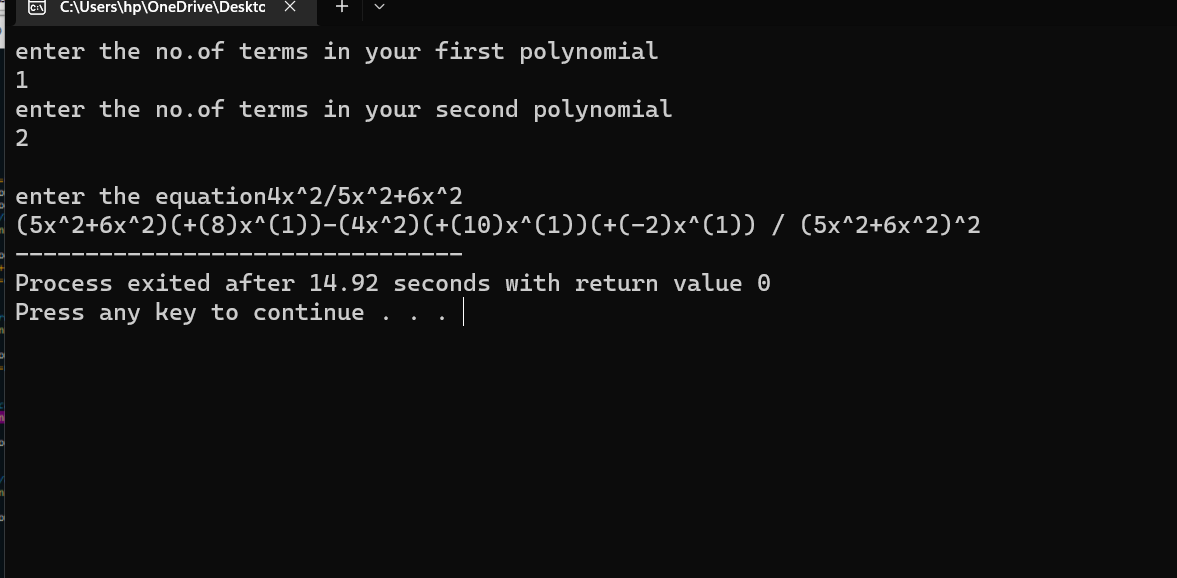
}

}

}



SAMPLE INPUT OUPUT FOR QUOTIENT RULE:



SAMPLE INPUT OUTPUT FOR LN AND EXPONENTIAL:

CODE:

#include<stdio.h>

#include<string.h>

int main()

{

char typ;

printf("please specify equation type:"

"\n type 'e' for exponential"

"\n type 'l' for natural log");

scanf(" %c",&typ);

switch(typ)

{

//NATURAL LOG

case 'l':

{

char eq[1000];

int n;

printf("enter the no.of terms of inside the brackets of ln\n");

scanf("%d",&n);

int coeff[n],pow[n],npow[n],ncoeff[n];

printf("enter the equation\n");

scanf("%s",eq);

int l=strlen(eq),j=0,k=3,x=6;

//extraction for coeff inside the brackets

for(int i=0;i<n;i++)

{

coeff[i]=(int)(eq[k]-'0');

k=k+5;

}

//extraction for powers inside the brackets

for(int i=0;i<n;i++)

{

pow[i]=(int)(eq[x]-'0');

x=x+5;

}

//CALCULATION

//calculation for coeff

for(int i=0;i<n;i++)

{

ncoeff[i]=coeff[i]\*pow[i];

}

//calculation for power

for(int i=0;i<n;i++)

{

npow[i]=pow[i]-1;

}

//PRINTING

for(int i=0;i<n;i++)

{

printf(" +(%d)x^%d",ncoeff[i],npow[i]);

}

printf("/");

for(int i=0;i<n;i++)

{

printf("+(%d)x^%d",coeff[i],pow[i]);

}

}

break;

//EXPONENTIAL

case 'e':

{

char eq[1000];

int n;

printf("enter the no.of terms of in the power of exponential\n");

scanf("%d",&n);

int coeff[n],pow[n],ncoeff[n],npow[n];

printf("enter the equation\n");

scanf("%s",eq);

int l=strlen(eq),j=0,k=3,x=6;

//extracrion for coefficents

for(int i=0;i<n;i++)

{

coeff[i]=(int)(eq[k]-'0');

k=k+5;

}

//extraction for power in the powerof exponential

for(int i=0;i<n;i++)

{

pow[i]=(int)(eq[x]-'0');

x=x+5;

}

//CALCULATION

//calculation for coeff

for(int i=0;i<n;i++)

{

ncoeff[i]=coeff[i]\*pow[i];

}

//calculation for power

for(int i=0;i<n;i++)

{

npow[i]=pow[i]-1;

}

//PRINTING

for(int i=0;i<n;i++)

{

printf("(+(%d)x^%d)x",ncoeff[i],npow[i]);

}

printf("e^");

for(int i=0;i<n;i++)

{

printf("(+(%d)x^%d)",coeff[i],pow[i]);

}

}

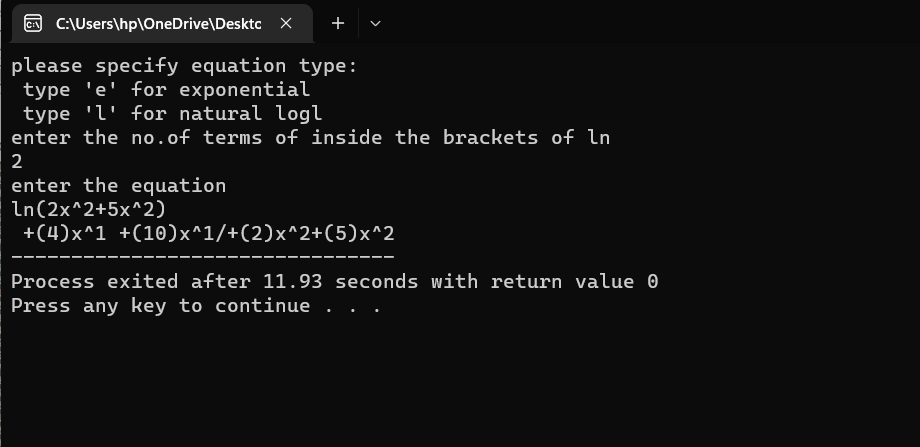
break;

}

return 0;

}

SAMPLE INPUT /OUTPUT FOR LN:



SAMPLE INPUT OUTPUT FOR EXXPONENTIAL:

Text

Description automatically generated

SAMPLE INPUT OUTPUT FOR TRIGNOMENTAREY:

CODE:

#include<stdio.h>

#include<string.h>

int main()

{

char eq[100];

int n,n1;

printf("enter the no.of terms in your trignometric equation\n");

scanf("%d",&n);

printf("\nenter the no.of terms inside the brackets of your trig function\n");

scanf("%d",&n1);

printf("\n enter the equation\n");

scanf("%s",eq);

int l=strlen(eq);

char sign;

for(int i=0;i<l;i++)

{

//SINE FUNCTION

if((eq[i]=='s')&&(eq[i+1]=='i')&&(eq[i+2]=='n'))

{

//extraction for sign

if(eq[i+9]=='\0')

{

sign='\0';

}

else

{

sign=eq[i+9];

}

int pow[n1],coeff[n1],ncoeff[n1],npow[n1],k=4+i,x=7+i;

//extraction for coeff

for(int i=0;i<n1;i++)

{

coeff[i]=(int)(eq[k]-'0');

k=k+5;

}

//extraction for power

for(int i=0;i<n1;i++)

{

pow[i]=(int)(eq[x]-'0');

x=x+5;

}

//calculation

for(int i=0;i<n1;i++)

{

ncoeff[i]=coeff[i]\*pow[i];

}

//calculation for power

for(int i=0;i<n1;i++)

{

npow[i]=pow[i]-1;

}

//PRINTING

for(int i=0;i<n1;i++)

{

printf("(+(%d)x^%d))",ncoeff[i],npow[i]);

}

printf("cos");

for(int i=0;i<n1;i++)

{

printf("((%d)x^%d)) %c ",coeff[i],pow[i],sign);

}

}

//END OF SINE FUNCTION

//COSINE FUNCTION

else if((eq[i]=='c')&&(eq[i+1]=='o')&&(eq[i+2]=='s'))

{

//extraction for sign

if(eq[i+9]=='\0')

{

sign='\0';

}

else

{

sign=eq[i+9];

}

int pow[n1],coeff[n1],ncoeff[n1],npow[n1],k=4+i,x=7+i;

//extraction for coeff

for(int i=0;i<n1;i++)

{

coeff[i]=(int)(eq[k]-'0');

k=k+5;

}

//extraction for power

for(int i=0;i<n1;i++)

{

pow[i]=(int)(eq[x]-'0');

x=x+5;

}

//calculation

for(int i=0;i<n1;i++)

{

ncoeff[i]=(coeff[i]\*pow[i])\*-1;

}

//calculation for power

for(int i=0;i<n1;i++)

{

npow[i]=pow[i]-1;

}

//PRINTING

for(int i=0;i<n1;i++)

{

printf("(+(%d)x^%d))",ncoeff[i],npow[i]);

}

printf("sin");

for(int i=0;i<n1;i++)

{

printf("(+(%d)x^%d)) %c ",coeff[i],pow[i],sign);

}

}

//TANGENT FUNCTION

else if((eq[i]=='t')&&(eq[i+1]=='a')&&(eq[i+2]=='n'))

{

//extraction for sign

if(eq[i+9]=='\0')

{

sign='\0';

}

else

{

sign=eq[i+9];

}

int pow[n1],coeff[n1],ncoeff[n1],npow[n1],k=4+i,x=7+i;

//extraction for coeff

for(int i=0;i<n1;i++)

{

coeff[i]=(int)(eq[k]-'0');

k=k+5;

}

//extraction for power

for(int i=0;i<n1;i++)

{

pow[i]=(int)(eq[x]-'0');

x=x+5;

}

//calculation

for(int i=0;i<n1;i++)

{

ncoeff[i]=(coeff[i]\*pow[i]);

}

//calculation for power

for(int i=0;i<n1;i++)

{

npow[i]=pow[i]-1;

}

//PRINTING

for(int i=0;i<n1;i++)

{

printf("(+(%d)x^%d))",ncoeff[i],npow[i]);

}

printf("(sec^2)");

for(int i=0;i<n1;i++)

{

printf("(+(%d)x^%d)) %c ",coeff[i],pow[i],sign);

}

}

}

return 0;

}

Text

Description automatically generated

SAMPLE INPUT OUTPUT FOR POLYNOMIAL:

CODE:

#include<stdio.h>

int main()

{

//inputs

int n,one,two,three;

printf("enter the no.of terms in your equation");

scanf("%d",&n);

int po[n];

int px[n];

printf("input the co-efficents of your terms in the polynomial");

for(int i=0;i<n;i++)

{

//coefficent input

scanf("%d",&px[i]);

}

printf("input the powers of your respective co-efficent of the term in the polynomial");

for(int i=0;i<n;i++)

{

//power input

scanf("%d",&po[i]);

}

//calculation

for(int i=0;i<n;i++)

{

if(po[i]==0)

{

one=two=0;

printf("+(%d)x^%d ",one,two);

}

else

{

one=px[i]\*po[i];

two=po[i]-1;

printf("+(%d)x^%d ",one,two);

}

}

}

Text

Description automatically generated