```
$ontext optimisation of ethylene cracker
$offtext

positive variables x1,x2,x3,x4,x5,x6,x7;
variables f,et,p,b,g,pro,feed,k ;
equations
profit,ethylene,propylene,butene,gasoil,procost,feedcost,ccap,pcap,ecap,prec,er
ec,thv;

profit.. f =e= pro - feed ;
ethylene..et =e= 17.75*((.5*x1) +(.5*x5)+(.35*x2)+(.35*x6)+(.2*x3)+(.25*x4));
propylene..p =e=
13.79*((0.01*x1)+(.01*X5)+(.15*x2)+(.15*x6)+(.15*x3)+(.18*x4));
butene..b =e= 26.64*((.01*x1)+(.01*x5)+(.02*x2)+(.02*x6)+(.04*x3)+(.05*x4));
gasoil.. g =e= 9.93*((.01*x1)+(.01*x5)+(.07*x2)+(.07*x6)+(.25*x3)+(.3*x4));
procost.. pro =e= et + p + b + g ;
feedcost.. feed =e= ((6.55*x1)+(9.73*x2)+(12.5*x3)+(10.14*x4));
ccap..((1.1*x1)+(.9*x2)+(.9*x3)+(1.0*x4)+(1.1*x5)+(.9*x6))=l=200000;
ecap..((0.5*x1)+(0.35*x2)+(0.25*x3)+(0.25*x4)+(0.5*x5)+(0.35*x6))=l=(100000);
pcap.. ((.01*x1)+(.15*x2)+(.15*x3)+(.18*x4)+(.01*x5)+(.15*x6))=l=(20000);
erec..x5 =e= ((.4*x1)+(.4*x5)+(.06*x2)+(.06*x6)+(.04*x3)+(.05*x4));
prec.. x6 =e= ((.1*x2)+(.1*x6)+(.01*x3)+(.01*x4));
thv..20000000=e=((-6857*x1)+(364*x2)+(2032*x3)-(1145*x4)-
(6857.6*x5)+(364*x6)+(21520*x7));

Retrieve(cracking.dat)
Retrieve(cracking.bas)
Retrieve(cracking.exe)
#include <stdio.h>
#include <conio.h>
#include<math.h>
#define N 10
#define NS 200
int m,nc=8;
double fun[N],dt[4],fs;
double x=0.0;
float f[N];
void main(void)
{

int i,j;
double w[N],coilrate,h,length,l0[N],l1[N],l2[N],l3[N],dumf[N];
double pw[N],pf[N], sum,conv,z;
float temp,frate,ws;
void eval(void);
clrscr();
```

```
dt[0]=0.635; dt[1]=0.698; dt[2]=0.762; dt[3]=0.826;

for(i=1;i<=nc;i++)
{ w[i]=0; f[i]=0; }

printf("Enter feedrate in kg/hr per zone \n");
scanf("%f",&frate);
coilrate=frate/8.0;

printf("Enter dilution steam flow rate in kg/hr per zone\n");
scanf("%f",&ws);
fs=(ws/144)*1000;

length=477.8;

printf("Enter composition of feedstock in wt%\n");
printf("Component\t\t Weight%\n");
printf("Methane\t\t\t");scanf("%f",&temp);w[4]=temp;
printf("Ethane\t\t\t");scanf("%f",&temp);w[1]=temp;
printf("Ethylene\t\t");scanf("%f",&temp);w[2]=temp;
printf("Propylene\t\t");scanf("%f",&temp);w[5]=temp;
printf("Propane\t\t\t");scanf("%f",&temp);w[6]=temp;

f[1]=(w[1]*coilrate*10)/30.0;
f[2]=(w[2]*coilrate*10)/28.0;
f[4]=(w[4]*coilrate*10)/16.0;
f[5]=(w[5]*coilrate*10)/42.0;
f[6]=(w[6]*coilrate*10)/44.0;


h=length/NS;
z=NS/4;

for(m=0;m<4;m++)
{
 for(j=0;j<z;j++)
 {


  for(i=1;i<=nc;i++)
   { dumf[i]=f[i];}

  eval();

  for(i=1;i<=nc;i++)
```

```
  { l0[i]=h*fun[i]; }

  x=x+h/2.0;

  for(i=1;i<=nc;i++)
   {f[i]=dumf[i]+l0[i]/2.0;}

  eval();

  for(i=1;i<=nc;i++)
   { l1[i]=h*fun[i]; }

  for(i=1;i<=nc;i++)
   {f[i]=dumf[i]+l1[i]/2.0;}

  eval();

  for(i=1;i<=nc;i++)
   { l2[i]=h*fun[i]; }

  x=x+h/2.0;

  for(i=1;i<=nc;i++)
   {f[i]=dumf[i]+l2[i];}

  eval();

  for(i=1;i<=nc;i++)
   { l3[i]=h*fun[i];
     f[i]=dumf[i]+(l0[i]+2*l1[i]+2*l2[i]+l3[i])/6.0;
   }
 }
}

sum=0.0;

pw[1]=30*f[1]; pw[2]=28*f[2];
pw[3]=2*f[3]; pw[4]=16*f[4];
pw[5]=42*f[5]; pw[6]=44*f[6];
pw[7]=26*f[7]; pw[8]=54*f[8];

for(i=1;i<=nc;i++)
{ sum=sum+pw[i];}

for(i=1;i<=nc;i++)
{ pf[i]=(pw[i]*100.0)/sum;}
```

```
printf("The Product Composition in wt%% is:\n");
printf("Component\t\t Weight%");
printf("\nHydrogen\t\t %.3f",pf[3]);
printf("\nMethane\t\t\t %.3f",pf[4]);
printf("\nAcetylene\t\t %.3f",pf[7]);
printf("\nEthylene\t\t %.3f",pf[2]);
printf("\nEthane\t\t\t %.3f",pf[1]);
printf("\nPropylene\t\t %.3f",pf[5]);
printf("\nPropane\t\t\t %.3f",pf[6]);
printf("\nButadiene\t\t %.3f",pf[8]);

conv=((w[1]-pf[1])*100)/w[1];


printf("\nThe percentage conversion is: %.3f",conv);

getche();
return;

}

void eval(void)
{
 int i;
 double r,rr,rt,prt,ft,kc1,kc5,kc4;
 double k1,k2,k5,k6,k8,k3,k4,k7,r1,r2,r3,r4,r5,r6,r8,r7,p,t,c;
 ft=0.0;
 t=969.81+1.05495*x-59.6595e-4*x*x+17.8821e-6*pow(x,3)-17.9787e-
9*pow(x,4);
 p=(3.29441-43.2254e-4*x+4.656734e-6*x*x-3.63109e-9*pow(x,3))/1.01325;

 r=0.0821;
 rr=1.987;
 rt=rr*t/1000.0;
 prt=p/(r*t);
 for(i=1;i<=nc;i++)
 { ft=ft+f[i]; }
 ft=ft+fs;
 c=M_PI*dt[m]*dt[m]*900;

 kc1=5.48e4*exp(-32.58/rt);
 kc5=1.91e4*exp(-30.16/rt);
 kc4=0.138;
 k1=4.652e13*exp(-65.20/(rt));
 k2=3.85e11*exp(-65.25/(rt));
```

```
 k3=4.692e10*exp(-50.6/(rt));
 k4=5.888e10*exp(-51.29/(rt));
 k5=9.814e8*exp(-36.92/(rt));
 k6=1.026e12*exp(-41.26/(rt));
 k7=1.514e11*exp(-55.8/rt);
 k8=7.083e13*exp(-60.43/(rt));

 r1=k1*((f[1]*prt/ft)-(f[2]*f[3]*prt*prt)/(ft*ft*kc1));
 r2=k2*(f[1]*prt/ft);
 r3=k3*(f[6]*prt/ft);
 r4=k4*(f[6]*prt/ft)-((f[5]*f[3]*prt*prt)/(ft*ft*kc4));
 r5=k5*((f[5]*prt)/ft-(f[7]*f[4]*prt*prt)/(ft*ft*kc5));
 r6=k6*((f[7]*f[2]*prt*prt)/(ft*ft));
 r7=k7*(f[5]*prt/ft);
 r8=k8*((f[1]*f[2]*prt*prt)/(ft*ft));

 fun[1]=-1*c*(r1+2*r2+r8);
 fun[2]=c*(r1-r6-r8+r3+3*r7);
 fun[3]=c*(r1+r4);
 fun[4]=c*(r2+r8+r5+r3);
 fun[5]=c*(r8-r5+r4-2*r7);
 fun[6]=c*(r2-r3-r4);
 fun[7]=c*(r5-r6);
 fun[8]=c*r6;
/* printf("k1=%.3f  k2=%.3f  k5=%.3f  k6=%.3f  k8=%.3f",k1,k2,k5,k6,k8);*/
 return;
}

 model cracking /all/;
 solve cracking using lp maximizing f;
 display f.l,x1.l,x2.l,x3.l;
```