```c
/* LINEAR NON-ISOTHERMAL RK 4TH ORDER METHOD */
#include<stdio.h>
#include<math.h>
void main()
{
long double T0,HR;
long double T,CB,k1,k2,k3;
int i;
long double A1,A2,A3,D1,D2,L1,L2,E3,R;
long double Cc1,CG1,Cc2,CG2,t;
long double n1,n2,n3,h;
long double a1,b1,c1,d1,e1;
long double a2,b2,c2,d2,e2;
long double a3,b3,c3,d3,e3;
long double a4,b4,c4,d4,e4;

/* INITIAL CONDITION */

CB=1.0;
Cc1=0.0;
CG1=0.0;
Cc2=0.0;
CG2=0.0;
/* INPUT */

printf(" Enter the value of initial temp(T0=773.0)\n");
scanf("%Lf",&T0);
printf("Enter the value of heating rate(HR)\n");
scanf("%Lf",&HR);
printf("Enter the value of order of reaction n1,n2,n3\n");
scanf("%Lf %Lf %Lf",&n1,&n2,&n3);
printf("Enter the value of time step h(t)\n");
scanf("%Lf",&h);
printf("\n");
printf("\n  h(t)       T         CB      CG1       Cc1       CG2     Cc2");
t=h;

/* RUNGE-KUTTA METHOD STARTS HERE */
while(1)
{
T=(HR)*t+T0;

A1=9.973*pow(10.0,-5.0);
A2=1.068*pow(10.0,-3.0);
A3=5.7*pow(10.0,5.0);
D1=17254.4;
D2=10224.4;
L1=-9061227.0;
L2=-6123081.0;
E3=81000.0;
R=8.314;
k1=A1*exp((D1/T)+(L1/(T*T)));
k2=A2*exp((D2/T)+(L2/(T*T)));
k3=A3*exp(-E3/(R*T));
a1=(-h)*((k1+k2)*pow(CB,n1));
b1=h*((k1*pow(CB,n1))-(k3*pow(CG1,n2)*pow(Cc1,n3)));
c1=h*((k2*pow(CB,n1))-(k3*pow(CG1,n2)*pow(Cc1,n3)));
d1=h*(k3*pow(CG1,n2)*pow(Cc1,n3));
e1=h*(k3*pow(CG1,n2)*pow(Cc1,n3));

a2=(-h)*((k1+k2)*pow((CB+(a1/2.0)),n1));
b2=h*((k1*pow((CB+(a1/2.0)),n1))-(k3*pow((CG1+(b1/2.0)),n2)*pow((Cc1+(c1/2.0)),n3)));
c2=h*((k2*pow((CB+(a1/2.0)),n1))-(k3*pow((CG1+(b1/2.0)),n2)*pow((Cc1+(c1/2.0)),n3)));
d2=h*(k3*pow((CG1+(b1/2.0)),n2)*pow((Cc1+(c1/2.0)),n3));
e2=h*(k3*pow((CG1+(b1/2.0)),n2)*pow((Cc1+(c1/2.0)),n3));
a3=(-h)*((k1+k2)*pow((CB+(a2/2.0)),n1));
b3=h*((k1*pow((CB+(a2/2.0)),n1))-(k3*pow((CG1+(b2/2.0)),n2)*pow((Cc1+(c2/2.0)),n3)));
c3=h*((k2*pow((CB+(a2/2.0)),n1))-(k3*pow((CG1+(b2/2.0)),n2)*pow((Cc1+(c2/2.0)),n3)));
d3=h*(k3*pow((CG1+(b2/2.0)),n2)*pow((Cc1+(c2/2.0)),n3));
e3=h*(k3*pow((CG1+(b2/2.0)),n2)*pow((Cc1+(c2/2.0)),n3));
a4=(-h)*((k1+k2)*pow((CB+a3),n1));
b4=h*((k1*pow((CB+a3),n1))-(k3*pow((CG1+b3),n2)*pow((Cc1+c3),n3)));
c4=h*((k2*pow((CB+a3),n1))-(k3*pow((CG1+b3),n2)*pow((Cc1+c3),n3)));
d4=h*(k3*pow((CG1+b3),n2)*pow((Cc1+c3),n3));
e4=h*(k3*pow((CG1+b3),n2)*pow((Cc1+c3),n3));
CB=CB+(1.0/6.0)*(a1+2.0*a2+2.0*a3+a4);
CG1=CG1+(1.0/6.0)*(b1+2.0*b2+2.0*b3+b4);
Cc1=Cc1+(1.0/6.0)*(c1+2.0*c2+2.0*c3+c4);
CG2=CG2+(1.0/6.0)*(d1+2.0*d2+2.0*d3+d4);
Cc2=Cc2+(1.0/6.0)*(e1+2.0*e2+2.0*e3+e4);
if(CB<=0.03)
break;


/* OUTPUT */

printf("\n%0.6Lf %0.6Lf %0.6Lf %0.6Lf %0.6Lf %0.6Lf %0.6Lf",t,T,CB,CG1,Cc1,CG2,Cc2);
t=t+h;
/*getch();*/
}
printf("\n");
getch();
}
```