```
   Help
#include  "bs1d_std.h"

static int CallSpread_BlackScholes_73(double s,
    double k1,double k2,double t,double r,double divid,
    double sigma,double *ptprice,double *ptdelta){
    double sigmasqrt,d1,d2,delta;

    sigmasqrt=sigma*sqrt(t);
    d1=(log(s/k1)+(r-divid)*t)/sigmasqrt+sigmasq
    rt/2.;
    d2=d1-sigmasqrt;
    delta=exp(-divid*t)*N(d1);

    *ptprice= s*delta -exp(-r*t)*k1*N(d2);
    *ptdelta=delta;

    d1=(log(s/k2)+(r-divid)*t)/sigmasqrt+sigmasq
    rt/2.;
    d2=d1-sigmasqrt;
    delta=exp(-divid*t)*N(d1);

  /*Price*/
    *ptprice-= s*delta -exp(-r*t)*k2*N(d2);

  /*Delta*/
    *ptdelta-=delta;

    return OK;
    }

int CALC(CF_CallSpread)(void *Opt,void *Mod,Pric
    ingMethod *Met)
    {
       TYPEOPT* ptOpt=(TYPEOPT*)Opt;
       TYPEMOD* ptMod=(TYPEMOD*)Mod;
       double r,divid;

       r=log(1.+ptMod->R.Val.V_DOUBLE/100.);
       divid=log(1.+ptMod->Divid.Val.V_DOUBLE/100
    .);
```

```
       return CallSpread_BlackScholes_73(ptMod->
    S0.Val.V_PDOUBLE,
           (ptOpt->PayOff.Val.V_NUMFUNC_1)->Par[0
    ].Val.V_PDOUBLE,(ptOpt->PayOff.Val.V_NUMFUNC_1)-
    >Par[1].Val.V_PDOUBLE,
          ptOpt->Maturity.Val.V_DATE-ptMod->T.
    Val.V_DATE,r,divid,ptMod->Sigma.Val.V_PDOUBLE,
           &(Met->Res[0].Val.V_DOUBLE),&(Met->Re
    s[1].Val.V_DOUBLE));
    }


int CHK_OPT(CF_CallSpread)(void *Opt, void *Mod)
    {
    return strcmp( ((Option*)Opt)->Name,"
    CallSpreadEuro");
    }

static int MET(Init)(PricingMethod *Met)
    {
    return OK;
    }

PricingMethod MET(CF_CallSpread)=
{
    "CF_CallSpread",
    {{" ",END,0,FORBID}},
    CALC(CF_CallSpread),
    {{"Price",DOUBLE,100,FORBID},{"Delta",DOUBLE,
    100,FORBID} ,{" ",END,0,FORBID}},
    CHK_OPT(CF_CallSpread),
    CHK_ok,
    MET(Init)
} ;
```

References