```
   Help
#include  "bs1d_std.h"


int Put_BlackScholes_73(double s,double k,double
    t,double r,double divid,double sigma,double *ptpr
    ice,double *ptdelta)
    {
    double sigmasqrt,d1,d2,delta;

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

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

  /*Delta*/
    *ptdelta=delta;

    return OK;
    }

int CALC(CF_Put)(void *Opt,void *Mod,Pricing
    Method *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 Put_BlackScholes_73(ptMod->S0.Val.V_PD
    OUBLE,
        (ptOpt->PayOff.Val.V_NUMFUNC_1)->Par[0].
    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->Res[1]
```

```
    .Val.V_DOUBLE));
    }


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

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

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

References