```
   Help
#include "doublim.h"
#include "error_msg.h"

extern char* path_sep;

int OPT(Get)(int user,Planning *pt_plan,Option *opt, Model
    *mod)
{
  TYPEOPT* pt=(  TYPEOPT*)(opt->TypeOpt);

  (opt->Init)(opt, mod);

  if (user==TOSCREEN)
    if  ((opt->Show)(user,pt_plan,opt,mod))
      do
  {
    Fprintf(TOSCREEN,"_____________________________Option:
    %s{n",opt->Name);

    ScanVar(pt_plan,user,&(pt->Maturity));
    GetParVar(pt_plan,user,(pt->PayOff.Val.V_NUMFUNC_1)->
    Par);

    /*if ((pt->RebOrNo).Val.V_BOOL==REBATE)*/
    GetParVar(pt_plan,user,(pt->Rebate.Val.V_NUMFUNC_1)->
    Par);

    GetParVar(pt_plan,user,(pt->LowerLimit.Val.V_NUMFUNC_1
    )->Par);
    GetParVar(pt_plan,user,(pt->UpperLimit.Val.V_NUMFUNC_1
    )->Par);
  }
      while ((opt->Show)(user,pt_plan,opt,mod));

  return (opt->Show)(TOSCREENANDFILE,pt_plan,opt,mod);
}
int OPT(FGet)(char **InputFile,int user,Planning *pt_plan,
    Option *opt, Model *mod)
{
  TYPEOPT* pt=(  TYPEOPT*)(opt->TypeOpt);
```

```
  (opt->Init)(opt, mod);

  if (user==TOSCREEN)
    {
      Fprintf(TOSCREEN,"_____________________________
    Option:%s{n",opt->Name);

      FScanVar(InputFile,pt_plan,user,&(pt->Maturity));
      FGetParVar(InputFile,pt_plan,user,(pt->PayOff.Val.V_
    NUMFUNC_1)->Par);

      /*if ((pt->RebOrNo).Val.V_BOOL==REBATE)*/
      FGetParVar(InputFile,pt_plan,user,(pt->Rebate.Val.V_
    NUMFUNC_1)->Par);

      FGetParVar(InputFile,pt_plan,user,(pt->LowerLimit.Val
    .V_NUMFUNC_1)->Par);
      FGetParVar(InputFile,pt_plan,user,(pt->UpperLimit.Val
    .V_NUMFUNC_1)->Par);
    }

  return (opt->Show)(TOSCREENANDFILE,pt_plan,opt,mod);
}

int OPT(Show)(int user,Planning *pt_plan,Option *opt,
    Model *mod)
{
  TYPEOPT* pt=(TYPEOPT*)(opt->TypeOpt);

  (opt->Init)(opt, mod);
  Fprintf(user,"##Option:%s{n",opt->Name);

  PrintVar(pt_plan,user,&(pt->Maturity));
  ShowParVar(pt_plan,user,(pt->PayOff.Val.V_NUMFUNC_1)->
    Par);

  /*if ((pt->RebOrNo).Val.V_BOOL==REBATE)*/
  ShowParVar(pt_plan,user,(pt->Rebate.Val.V_NUMFUNC_1)->
    Par);
```

```
  ShowParVar(pt_plan,user,(pt->LowerLimit.Val.V_NUMFUNC_1)-
    >Par);
  ShowParVar(pt_plan,user,(pt->UpperLimit.Val.V_NUMFUNC_1)-
    >Par);

  return (opt->Check)(user,pt_plan,opt);
}



extern Option OPT(DoubleCallOutEuro);
extern Option OPT(DoublePutInEuro);
extern Option OPT(DoublePutOutEuro);
extern Option OPT(DoubleCallInEuro);
extern Option OPT(DoubleCallOutAmer);
extern Option OPT(DoublePutInAmer);
extern Option OPT(DoublePutOutAmer);
extern Option OPT(DoubleCallInAmer);
extern Option OPT(ParisianDoubleCallOutEuro);
extern Option OPT(ParisianDoubleCallInEuro);

Option* OPT(family)[]={
  &OPT(DoubleCallOutEuro),
  &OPT(DoublePutOutEuro),
  &OPT(DoubleCallInEuro),
  &OPT(DoublePutInEuro),
  &OPT(DoubleCallOutAmer),
  &OPT(DoublePutOutAmer),
  &OPT(DoubleCallInAmer),
  &OPT(DoublePutInAmer),
  &OPT(ParisianDoubleCallOutEuro),
  &OPT(ParisianDoubleCallInEuro),
  NULL,
};
```

References