```
   Help
#include "doublim.h"

extern char* path_sep;
extern char **error_msg;

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

    (opt->Init)(opt);

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

                ScanVar(pt_plan,user,&(pt->Maturi
    ty));
                GetParVar(pt_plan,user,(pt->PayOf
    f.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->Low
    erLimit.Val.V_NUMFUNC_1)->Par);
                GetParVar(pt_plan,user,(pt->Upp
    erLimit.Val.V_NUMFUNC_1)->Par);
            }
            while ((opt->Show)(user,pt_plan,opt))
    ;

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

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

  (opt->Init)(opt);
    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);
    }

int OPT(Check)(int user,Planning *pt_plan,Option
    *opt)
    {
    TYPEOPT* pt=(TYPEOPT*)(opt->TypeOpt);
    int status=OK;
    char helpfile[MAX_PATH_LEN]="";

  if ((strlen(opt->Name)+strlen(opt->ID)+strlen(
    "{{opt{{") +strlen("{{")
    +strlen("_doc.pdf"))>=MAX_PATH_LEN)
  {
    Fprintf(TOSCREEN,"%s{n",error_msg[PATH_TOO_
    LONG]);
        exit(WRONG);
  }
```

```
  strcpy(helpfile,path_sep);
  strcat(helpfile,"opt");
  strcat(helpfile,path_sep);

  strcat(helpfile,opt->ID);
  strcat(helpfile,path_sep);

  strcat(helpfile,opt->Name);
  strcat(helpfile,"_doc.pdf");

    status+=ChkVar(pt_plan,&(pt->Maturity));
    status+=((pt->PayOff.Val.V_NUMFUNC_1)->Check)
    (user,pt_plan,pt->PayOff.Val.V_NUMFUNC_1);

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

    status+=((pt->LowerLimit.Val.V_NUMFUNC_1)->
    Check)(user,pt_plan,pt->LowerLimit.Val.V_NUMFUNC_1
    );
    status+=((pt->UpperLimit.Val.V_NUMFUNC_1)->
    Check)(user,pt_plan,pt->UpperLimit.Val.V_NUMFUNC_1
    );

    return Valid(user,status,helpfile);
    }

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