```
   Help
#include "lim.h"

extern char* path_sep;
extern char **error_msg;

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

  (void)(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->Maturity));
          GetParVar(pt_plan,user,(pt->Limit.Val.
    V_NUMFUNC_1)->Par);
          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);
        }
      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);

  (void)(opt->Init)(opt);
```

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

  PrintVar(pt_plan,user,&(pt->Maturity));
  ShowParVar(pt_plan,user,(pt->Limit.Val.V_
    NUMFUNC_1)->Par);
  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);

  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->Limit.Val.V_NUMFUNC_1)->Check)(u
    ser,pt_plan,pt->Limit.Val.V_NUMFUNC_1);
    status+=((pt->PayOff.Val.V_NUMFUNC_1)->Check)
    (user,pt_plan,pt->PayOff.Val.V_NUMFUNC_1);
    status+=((pt->Rebate.Val.V_NUMFUNC_1)->Check)
    (user,pt_plan,pt->Rebate.Val.V_NUMFUNC_1);

    return Valid(user,status,helpfile);
}

extern Option OPT(PutUpOutEuro);
extern Option OPT(CallDownOutAmer);
extern Option OPT(CallDownOutEuro);
extern Option OPT(CallUpInEuro);
extern Option OPT(CallUpInAmer);
extern Option OPT(CallUpOutAmer);
extern Option OPT(CallUpOutEuro);
extern Option OPT(PutDownInEuro);
extern Option OPT(PutDownInAmer);
extern Option OPT(PutDownOutAmer);
extern Option OPT(PutDownOutEuro);
extern Option OPT(PutUpInEuro);
extern Option OPT(PutUpInAmer);
extern Option OPT(PutUpOutAmer);
extern Option OPT(CallDownInEuro);
extern Option OPT(CallDownInAmer);
extern Option OPT(ParisianCallDownOutEuro);
extern Option OPT(ParisianCallDownInEuro);

Option* OPT(family)[]={
        &OPT(CallDownOutEuro),
        &OPT(PutDownOutEuro),
        &OPT(CallUpOutEuro),
        &OPT(PutUpOutEuro),
        &OPT(CallDownInEuro),
        &OPT(CallUpInEuro),
        &OPT(PutDownInEuro),
        &OPT(PutUpInEuro),
        &OPT(CallDownOutAmer),
        &OPT(PutDownOutAmer),
        &OPT(CallUpOutAmer),
```

```
        &OPT(PutUpOutAmer),
        &OPT(CallDownInAmer),
        &OPT(PutDownInAmer),
        &OPT(CallUpInAmer),
        &OPT(PutUpInAmer),
    &OPT(ParisianCallDownOutEuro),
    &OPT(ParisianCallDownInEuro),
        NULL
    };
```

References