

```

    Help
#ifndef _OPTYTYPE_H
#define _OPTYTYPE_H

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <malloc.h>
#include <string.h>
#include <stdarg.h>
#include <time.h>
#include <ctype.h>

#ifdef _WIN32
#include <process.h> /*For calling Acrobat for
    help file*/
#else
#define _spawnlp Spawnlp
#define _P_WAIT 0
#define _MAX_PATH 248
#endif

#include "error_msg.h"
#define MC 0 /*dans optype.h plutot*/
#define QMC 1 /*id*/
#define MONTECARLOMAX 10000 /*id*/
#define CREATE 0 /*id*/
#define RETRIEVE 1 /*id*/
#define GEN_NUMBER 20

/*-----MACROS-----
-----*/

#define TOSTR(X) #X
#define TOSTR_2(X) TOSTR(X) /*if X is a macro,
    this forces evaluation*/
#define MERGE2_2(X,Y) MERGE2(X,Y)
#define MERGE2(X,Y) X##_##Y
#define MERGE3_2(X,Y,Z) MERGE3(X,Y,Z)

```

```
#define MERGE3(X,Y,Z) X##_##Y##_##Z
#define MERGE4_2(X,Y,Z,T) MERGE4(X,Y,Z,T)
#define MERGE4(X,Y,Z,T) X##_##Y##_##Z##_##T
#define MERGE5_2(X,Y,Z,T,U) MERGE5(X,Y,Z,T,U)
#define MERGE5(X,Y,Z,T,U) X##_##Y##_##Z##_##T##_##
    #U
```

```
/*-----CONST&TYPE
    S-----*/
```

```
#define MAX_PATH_LEN _MAX_PATH
#define MAX_CHAR 80
#define MAX_CHAR_X3 240
#define MAX_CHAR_X4 320
#define MAX_MET 40
#define MAX_OPT 30
#define MAX_PAR 30
#define MAX_METHODS 40 /* = max number of Pricing
    methods */
```

```
typedef char Label[MAX_CHAR];
```

```
#define NO_PAR -1
#define OK 0
#define WRONG 1
#define UNABLETOOPENFILE 2
#define MEMORYALLOCATIONERROR 3
#define DONOTITERATE 16
#define TOSCREEN 0
#define TOFILE 1
#define TOSCREENANDFILE 2
#define NAMEONLYTOFILE 3
#define VALUEONLYTOFILE 4
```

```
#define ZOOMTIME 1000
```

```
/*-----VAR-----
    -----*/
```

```

#define MAX_ITERATOR 3

typedef struct VAR{
    Label    Vname;
    int Vtype;
    union    {
        int V_INT;
        int V_INT2;
        double V_DOUBLE;
        long V_LONG;
        double V_PDOUBLE;
        double V_SPDOUBLE;
        double V_RGDOUBLE051;
        double V_DATE;
        double V_RGDOUBLE;
        double V_RGDOUBLEM11;
        double V_RGDOUBLE12;
        int V_BOOL;
        int V_PADE;
        int V_RGINT13;
        int V_GENER;
        double V_RGDOUBLE14;
        struct NumFunc_1* V_NUMFUNC_1;
        struct NumFunc_2* V_NUMFUNC_2;
        struct PtVar* V_PTVAR;
        struct DoubleArray* V_DOUBLEARRAY;
    } Val;
    int Viter;
} VAR;          /* typedef struct A{ }A; is equivalent to: struct A{ };typedef struct A A;*/

/*Vtype*/
#define FIRSTLEVEL 20

/*FirstClass*/
#define END 0
#define INT 1
#define DOUBLE 2
#define LONG 3
#define PDOUBLE 4

```

```
#define DATE 5
#define RGDOUBLE 6
#define BOOL 7
#define PADE 8
#define RGDOUBLE12 9
#define INT2 10
#define RGINT13 11
#define SPDOUBLE 12
#define RGDOUBLE051 13
#define GENER 14
#define RGDOUBLE14 15
#define RGDOUBLEM11 16

/*SecondClass*/

#define NUMFUNC_1 20
#define NUMFUNC_2 21
#define PTVAR 22
#define DOUBLEARRAY 23
/*This last should be less than MAX_TYPE:*/

#define MAX_TYPE 30
/*Viter*/
#define IRRELEVANT -3
#define FORBID -2
#define ALLOW -1
#define ALREADYITERATED 256
/*MAX_ITERATOR should be less than ALREADYITERA
    TED*/

/*Useful Flags*/
#define EURO 0
#define AMER 1
#define TOTAL 0
#define PARTIAL 1
#define CONT 0
#define DISC 1
#define OUT 0
#define IN 1
#define DOWN 0
#define UP 1
```

```

#define REBATE 0
#define NOREBATE 1
#define CONSTLIM 0
#define MOVLIM 1
#define TIMEAVERAGING 10

/*-----PLANNING-----
   -----*/

#define MAX_ITER 1000

typedef struct Iterator{
    VAR*    Location;
    VAR     Min;
    VAR     Max;
    VAR     Default;
    int     StepNumber;
} Iterator;

typedef struct Planning{
    Iterator  Par[MAX_ITERATOR];
    int       VarNumber;
    char      Action;
    int       NumberOfMethods;
} Planning;

/*SecondLevelVars*/
/*Arrays of VAR*/

typedef struct PtVar{
    VAR Par[MAX_PAR];
} PtVar;

/*NumericalFunctions*/

typedef struct NumFunc_1{
    double      (*Compute)(VAR*,double);
    VAR Par[MAX_PAR];
    int         (*Check)(int user,Planni
        ng*,void*);
} NumFunc_1;

```

```

typedef struct NumFunc_2{
    double          (*Compute)(VAR*,double,double)
    ;
    VAR Par[MAX_PAR];
    int              (*Check)(int user,Planni
        ng*,void*);
} NumFunc_2;

/*Arrays of Double*/
typedef struct DoubleArray{
    long size;
    double *array;
}DoubleArray;

/*-----MODELS-----
-----*/

typedef struct Model{
    Label      ID;
    Label      Name;
    void*      TypeModel;
    int        (*Get)(int user, Planning*,str
        uct Model*);
    int        (*Show)(int user,Planning*,str
        uct Model*);
    int        (*Check)(int user,Planning*,
        struct Model*);
    int        (*Init)(struct Model*);
} Model;

#define MOD(X) MERGE2_2(TYPEMOD,X)
#define MAKEMOD(X)  MAKEMODEL(TOSTR_2(TYPEMOD),##
    X)
#define MAKEMODEL(Z,X) Model MOD(model)={Z ,#X,&#
    #X,MOD(Get),MOD(Show),MOD(Check),MOD(Init)}

/*-----OPTIONS-----
-----*/

typedef struct Option{

```

```

Label      ID;
Label      Name;
void*      TypeOpt;
int        (*Get)(int user, Planning*,str
            uct Option*);
int        (*Show)(int user,Planning*,str
            uct Option*);
int        (*Check)(int user,Planning*,
            struct Option*);
int        (*Init)(struct Option*);
} Option;

#define OPT(X) MERGE2_2(TYPEOPT,X)
#define MAKEOPT(X) MAKEOPTION(TOSTR_2(TYPEOPT),#
            #X)
#define MAKEOPTION(Z,X) Option OPT(##X)={Z ,#X,&#
            #X,OPT(Get),OPT>Show),OPT(Check),OPT(Init)}

typedef Option* Family[MAX_OPT];

/*-----PRICINGS &
   DYNAMIC TESTS-----
   */

/*Pricing Methods*/
typedef struct PricingMethod{
    Label      Name;
    VAR Par[MAX_PAR];
    int        (*Compute)
        (void*,void*,struct PricingMethod*);
    VAR Res[MAX_PAR];
    int        (*Check
        Opt)(void*,void*);
    int        (*Check)(
        int user, Planning*,void*);
    int        (*Init)(
        struct PricingMethod*);
} PricingMethod;

#define MET(X) MERGE3_2(TYPEMOD,TYPEOPT,X)
#define CALC(X) MERGE4_2(CALC,TYPEMOD,TYPEOPT,X)

```



```
VAR Res[MAX_PAR];  
int (*Check)(  
    int user, Planning *, struct TimeInfo *);  
int (*Init)(  
    struct TimeInfo*);  
} TimeInfo;  
  
#endif
```

## References