Help

```
/*
 * Writen by David Pommier <david.pommier@gmail.com>
 * INRIA 2009
 */

#include "
href../../../common/math/equity_pricer/gd_list_h_src.pdfgd_list.h"

/**
 * allocates a contains.
 * @param ind key
 * @param val value
 * @return a pointeur to PremiaContains
 */
PremiaContains *premia_contains_create(const int ind, double val)
{
  PremiaContains *C;
  if ((C = malloc(sizeof(PremiaContains))) == NULL) return NULL;
  C->index = ind;
  C->value = val;
  return C;
}

PremiaContains *premia_contains_clone(int ind, double val)
{
  PremiaContains *C;
  if ((C = malloc(sizeof(PremiaContains))) == NULL) return NULL;
  C->index = ind;
  C->value = val;
  return C;
}

/**
 * allocates a contains - copy constructor.
 * @param C2 contains pointer
 * @return a pointeur to PremiaContains
 */
PremiaContains *premia_contains_copy(const PremiaContains *C2)
{
```

```
  PremiaContains *C;
  if ((C = malloc(sizeof(PremiaContains))) == NULL) return NULL;
  C->index = C2->index;
  C->value = C2->value;
  return C;
}

/**
 * free a contains
 * @param C address of a contains
 */
void premia_contains_free(PremiaContains **C)
{
  if (*C != NULL)
    {
      free(*C);
      *C = NULL;
    }
}

/**
 * Prints a contains to a file
 *
 * @param fic a file descriptor.
 * @param C a Contians pointer.
 */
void premia_contains_fprint(FILE *fic, PremiaContains *C)
{
  fprintf(fic, " ( %d,  %7.4f) ;", C->index, C->value);
}


/**
 * Add at value, the value of PremiaContains C2
 *
 * @param C a PremiaContains pointer, C.Value+ =C2 .Value.
 * @param C2 a Contians pointer.
 */
void  premia_contains_add(PremiaContains *C, const PremiaContains *C2)
{
  C->value += C2->value;
```

```
}

/**
 * Less compute relation C1<C2
 *
 * @param C1 a PremiaContains pointer.
 * @param C2 a Contians pointer.
 * @return a int C1<C2
 */
int  premia_contains_less(const PremiaContains *C1, const PremiaContains *C2)
{
  return C1->index < C2->index;
}

/**
 * Equal compute relation C1==C2
 *
 * @param C1 a PremiaContains pointer.
 * @param C2 a Contians pointer.
 * @return a int C1==C2
 */
int  premia_contains_equal(const PremiaContains *C1, const PremiaContains *C2)
{
  return C1->index == C2->index;
}
```