Help

```
#include  "hes1d_pad.h"

int MOD_OPT(ChkMix)(Option *Opt, Model *Mod)
{
  TYPEOPT *ptOpt = (TYPEOPT *)(Opt->TypeOpt);
  TYPEMOD *ptMod = (TYPEMOD *)(Mod->TypeModel);
  int status = OK;

  if (ptOpt->Maturity.Val.V_DATE <= ptMod->T.Val.V_DATE)
    {
      Fprintf(TOSCREENANDFILE, "Current date greater than maturity!\ n");
      status += 1;
    };
  if ((ptOpt->MinOrElse).Val.V_BOOL == MINIMUM)
    {
      if ((ptOpt->PathDep.Val.V_NUMFUNC_2)->Par[4].Val.V_PDOUBLE > ptMod->S0.Val
        {
          Fprintf(TOSCREENANDFILE, "Minimum greater than spot!\ n");
          status += 1;
        };
    }
  if ((ptOpt->MinOrElse).Val.V_BOOL == MAXIMUM)
    {
      if ((ptOpt->PathDep.Val.V_NUMFUNC_2)->Par[4].Val.V_PDOUBLE < ptMod->S0.Val
        {
          Fprintf(TOSCREENANDFILE, "Maximum lower than spot!\ n");
          status += 1;
        };
    }
  return status;
}


extern PricingMethod MET(MC_AsianKusuoka_Heston);
extern PricingMethod MET(MC_AsianKNN_Heston);
extern PricingMethod MET(MC_AsianFunctionalQuantization_Heston);
extern PricingMethod MET(MC_AsianAlfonsi_Heston);
extern PricingMethod MET(MC_AsianKusuokaEuler_Heston);
```

```
extern PricingMethod MET(MC_Am_Asian_Alfonsi_LongstaffSchwartz_hes1d);
extern PricingMethod MET(MC_Am_Asian_Alfonsi_AndersenBroadie_hes1d);
extern PricingMethod MET(AP_FJM_ASIAN_HESTON);

PricingMethod *MOD_OPT(methods)[] =
{
  &MET(MC_AsianKNN_Heston),
  &MET(MC_AsianFunctionalQuantization_Heston),
  &MET(MC_AsianAlfonsi_Heston),
  &MET(MC_AsianKusuoka_Heston),
  &MET(MC_AsianKusuokaEuler_Heston),
  &MET(MC_Am_Asian_Alfonsi_LongstaffSchwartz_hes1d),
  &MET(MC_Am_Asian_Alfonsi_AndersenBroadie_hes1d),
  &MET(AP_FJM_ASIAN_HESTON),
  NULL
};

DynamicTest *MOD_OPT(tests)[] =
{
  NULL
};

Pricing MOD_OPT(pricing) =
{
  ID_MOD_OPT,
  MOD_OPT(methods),
  MOD_OPT(tests),
  MOD_OPT(ChkMix)
};
```