Help

```
#include  "hes1d_vol.h"

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

  if ((strcmp(Opt->Name, "VarianceSwap") == 0) || (strcmp(Opt->Name, "Volatility
    if ((ptOpt->Maturity.Val.V_DATE) <= (ptMod->T.Val.V_DATE))
      {
        Fprintf(TOSCREENANDFILE, "Current date greater than maturity!\ n");
        status += 1;
      };

  return status;
}

extern PricingMethod MET(AP_HES_VS_ZHOU);
extern PricingMethod MET(AP_HES_REALVAR);
extern PricingMethod MET(AP_HES_VARIANCESWAP);
extern PricingMethod MET(CF_HES_VARIANCESWAP);
extern PricingMethod MET(AP_HESCHIARELLA_VARIANCESWAP);
extern PricingMethod MET(AP_HES_VOLATILITYSWAP);
extern PricingMethod MET(AP_HES_VOLATILITYSWAP2);
extern PricingMethod MET(MC_Timer);

PricingMethod *MOD_OPT(methods)[] =
{
  &MET(AP_HES_VS_ZHOU),
  &MET(AP_HES_REALVAR),
  &MET(CF_HES_VARIANCESWAP),
  &MET(AP_HESCHIARELLA_VARIANCESWAP),
  &MET(AP_HES_VARIANCESWAP),
  &MET(AP_HES_VOLATILITYSWAP),
  &MET(AP_HES_VOLATILITYSWAP2),
  &MET(MC_Timer),
  NULL
};
```

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

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