Help

```
#include "optype.h"
#include "var.h"
#include "chk.h" /*Because of the object at end of file*/
#include "timeinfo.h"
#include "error_msg.h"

extern char premiasrcdir[MAX_PATH_LEN];
extern char premiamandir[MAX_PATH_LEN];
extern char *path_sep;

int Chk_TimeInfo_OK(int user, Planning *pt_plan, TimeInfo *Met)
{
  return OK;
}

/*_____________________________________TIMEINFO_____________________________________
int   GetTimeInfo(int user, Planning *pt_plan, TimeInfo *Met)
{
  char helpfile[MAX_PATH_LEN] = "";

  if ((2 * strlen(path_sep) + strlen("common")
       + strlen("timeinfo_src.pdf")) >= MAX_PATH_LEN)
    {
      Fprintf(TOSCREEN, "%s\ n", error_msg[PATH_TOO_LONG]);
      exit(WRONG);
    }

  strcpy(helpfile, premiamandir);
  strcpy(helpfile, path_sep);
  strcat(helpfile, "common");
  strcat(helpfile, path_sep);
  strcat(helpfile, "timeinfo_src.pdf");

  if (pt_plan->Action == 'p')
    {
      (Met->Init)(Met);

      if (user == TOSCREEN)
        {
```

```
          Fprintf(TOSCREEN, "\ n%s", Met->Name);

          if (Valid(TOSCREEN, OK, helpfile) == WRONG)
            {
              Met->Par[0].Val.V_INT = OK;

              if (ShowTimeInfo(user, pt_plan, Met))
                {
                  do
                    {
                      GetParVar(pt_plan, user, Met->Par + 1);
                    }
                  while (ShowTimeInfo(user, pt_plan, Met));
                }

              return ShowTimeInfo(TOSCREENANDFILE, pt_plan, Met);
            }
          else
            {
              Met->Par[0].Val.V_INT = WRONG;
              return OK;
            }

        }

      return ShowTimeInfo(TOSCREENANDFILE, pt_plan, Met);
    }
  else
    return OK;
}


int ShowTimeInfo(int user, Planning *pt_plan, TimeInfo *Met)
{
  char helpfile[MAX_PATH_LEN] = "";

  if (pt_plan->Action == 'p')
    {

      if (Met->Par[0].Val.V_INT == WRONG)
        {
```

```
          return OK;
        }
      else
        {
          if (user == TOSCREENANDFILE)
            {
              Fprintf(TOSCREEN, "\ n\ n##TimeInfo:%s\ n", Met->Name); /*TOSCREEN
              ShowParVar(pt_plan, user, Met->Par + 1);

            }
          else
            {
              if (ShowParVar(pt_plan, user, Met->Par + 1) == OK)
                {
                  return Valid(user, ChkParVar(pt_plan, Met->Par + 1), helpfile)
                }
              else
                return Valid(NO_PAR, ChkParVar(pt_plan, Met->Par + 1), helpfile)
            }

        }

    }
  return OK;
}

int ShowResultTimeInfo(int user, Planning *pt_plan, int error, TimeInfo *Met)
{
  if ((pt_plan->Action == 't') || (Met->Par[0].Val.V_INT == WRONG))
    {
      return OK;
    }
  else
    {
      if ((error == 0) || (user == NAMEONLYTOFILE))
        {
          ShowParVar(pt_plan, user, Met->Res);
        }
      else
        {
          Fprintf(user, "%s\ n", error_msg[error - 1]);
```

```
        }

      return OK;
    }
}


static int Init(TimeInfo  *Met)
{
  static int first = 1;

  if (first)
    {
      Met->Par[1].Vtype = INT;
      Met->Par[1].Val.V_INT = 10;
      Met->Par[1].Viter = ALLOW;

      Met->Par[2].Vtype = LONG;
      Met->Par[2].Val.V_LONG = 1;
      Met->Par[2].Viter = ALLOW;


      Met->Par[0].Vtype = INT;
      Met->Par[0].Val.V_INT = OK;
      Met->Par[0].Viter = ALLOW;

      Met->Res[0].Vtype = DOUBLE;

      first = 0;
    }


  return OK;
}

TimeInfo computation_time_info =
{
  "No Computation Time Information",
  { {"Choice", INT, {100}, FORBID},
    {"AveragingTimeWidth", INT, {100}, FORBID},
    {"NumberOfRuns", LONG, {100}, FORBID},
```

```
    {" ", PREMIA_NULLTYPE, {0}, FORBID}
  },
  { {"MeanTime(ms)", DOUBLE, {100}, FORBID},
    {" ", PREMIA_NULLTYPE, {0}, FORBID}
  },
  Chk_TimeInfo_OK,
  Init
} ;
```