02 oct 00 10:04 **makefile** Page 1/1

```
CC = gcc                      #nom du compilateur
CFLAGS = -Wall -pedantic -g #options de la compilation
LDFLAGS  =                    #options du linker


convertir_en: convertir_en.o devise.o

convertir_en.o: convertir_en.c devise.h

devise.o:  devise.h devise.c

clean:
        rm -f *.o *~
```


Imprimé par Pierre−Andre WACRENIER

10 nov 00 10:20 **devise.h** Page 1/1

```
#ifndef DEVISE_H
#define DEVISE_H

typedef enum
  {
    Euro = 0, Franc, Deutschmark, Devise_inconnue
  }
devise;

extern double dev_convertir (devise origine, devise cible, double montant);

extern devise dev_reconnaitre (char *chaine);

#endif
```


Imprimé par Pierre−Andre WACRENIER

02 oct 00 15:27 **convertir_en.c** Page 1/1

```
#include <stdlib.h>
#include <stdio.h>
#include "devise.h"

#define NOMBRE_ARGUMENTS 4

void usage (char *commande);

int
main (int argc, char *argv[])
{
  devise origine, cible;

  if (argc != NOMBRE_ARGUMENTS)
    usage (argv[0]);

  cible = dev_reconnaitre (argv[1]);
  origine = dev_reconnaitre (argv[3]);
  if (origine == Devise_inconnue || cible == Devise_inconnue)
    usage (argv[0]);

  printf ("%f\n", dev_convertir (origine, cible, atof (argv[2])));

  return EXIT_SUCCESS;
}

void
usage (char *commande)
{
  printf ("usage %s nom_de_devise_cible "
          "montant nom_de_devise_origine \n", commande);
  exit (EXIT_FAILURE);
}
```


Imprimé par Pierre−Andre WACRENIER

07 oct 01 17:26 **devise.c** Page 1/1

```
#include "devise.h"
#include <assert.h>
#include <stdlib.h>
#include <string.h>

const char *nom_de_devise[] =
{"euro", "franc", "deutschmark"};

static double valeur_en_euro[] =
{1,                             /* euro */
 0.15245,                       /* franc */
 0.51129                        /* deutschmark */
};

double
dev_convertir (devise origine, devise cible, double montant)
{
  assert (origine < Devise_inconnue && cible < Devise_inconnue);
  return montant * valeur_en_euro[origine] / valeur_en_euro[cible];
}

devise
dev_reconnaitre (char *chaine)
{
  devise d;
  assert (chaine != NULL);

  for (d = Euro; d < Devise_inconnue; d++)
    if (!strcmp (chaine, nom_de_devise[d]))
      return d;

  return Devise_inconnue;
}
```