Nom/Prénom :

1 Questions courtes

1.1 Makefile

1. Soit le `makefile` suivant

```
LIBDIR=../../lib
INCDIR=../../inc

CC=gcc
CCFLAGS=-I${INCDIR} -c
LDFLAGS=-L${LIBDIR} -llist

TARGET=myProg

OBJECTS=${TARGET}.o part1.o part2.o

${TARGET} : ${OBJECTS}
	${CC} -o ${TARGET} ${OBJECTS} ${LDFLAGS}

.c.o :
	${CC} ${CCFLAGS} $<

part2.o : part2.c ${INCDIR}/myHeader.h

part1.o : part1.c ${INCDIR}/myHeader.h

${TARGET}.o : ${TARGET}.c ${INCDIR}/myHeader.h

clean :
	rm -f ${OBJECTS} ${TARGET}
```

note : dans un makefile, les caractères {} sont équivalents aux caractères ().

2. Quelle est l'intention de l'utilisateur lorsqu'il entre la commande suivante : `$ make` (le `$` représente le prompt du shell).

3. Afficher les commandes successives qui sont automatiquement lancées lorsque l'utilisateur tape la commande : `$ make`.

4. Expliquer le role de l'option `-I$INCDIR`

5. Expliquer le role de l'option `-llist`

Nom/Prénom :

1.2 Pointeurs

1. Compléter la fonction suivante pour qu'elle renvoie une COPIE de la chaine de caractères passée en argument.

```
char * duplicateString(const char * str)
   {



   }
```

2. La fonction suivante est incorrecte.

```
int findOption(const char * opt)
   {
   if(opt=="-ra") return(1);
   if(opt=="-rb") return(2);
   if(opt=="-wa") return(3);
   if(opt=="-wb") return(4);
   return(0);
  }
```

Pourquoi ?

Proposez une correction

3. Quel est le sens de la notation suivante : `float* (*calculer)(float*)` ?

Nom/Prénom :

2 Structure Shadock

Vous allez participer au développement d'une application particulière concernant les Shadocks. Comme chacun sait, un shadock pompe, et c'est tout. Pour cela, il utilise une pompe. Le fichier d'entête des shadocks est donc celui-ci :

```
#ifndef SHADOCK_H
#define SHADOCK_H

typedef struct {
  int brasGauche;
  int brasDroit;
  char* petitNom;
} Shadock;


typedef struct {
  float quantitee;
  char* numSerie;
} PompeAire;


void pompeADroite(Shadock *unShadock, PompeAire *unePompe);
void pompeAGauche(Shadock *unShadock, PompeAire *unePompe);
void afficheUnShadock( Shadock *unShadock);
void affichePompeAire(PompeAire *unePompe);

#endif
```

Le programme que vous devez écrire doit mettre en jeu deux Shadocks, dont les noms sont passés en ligne de commande, ainsi qu'une pompe dont on donne le numéro de série, le dernier argument est le nombre de "pompages" que doit effectuer chacun des Shadocks.

En annexe 1 deux exemples d'exécution du programme à développer sont donnés. En annexe 2, les codes des différentes fonctions sont fournis.
Fonctionnement de l'application :

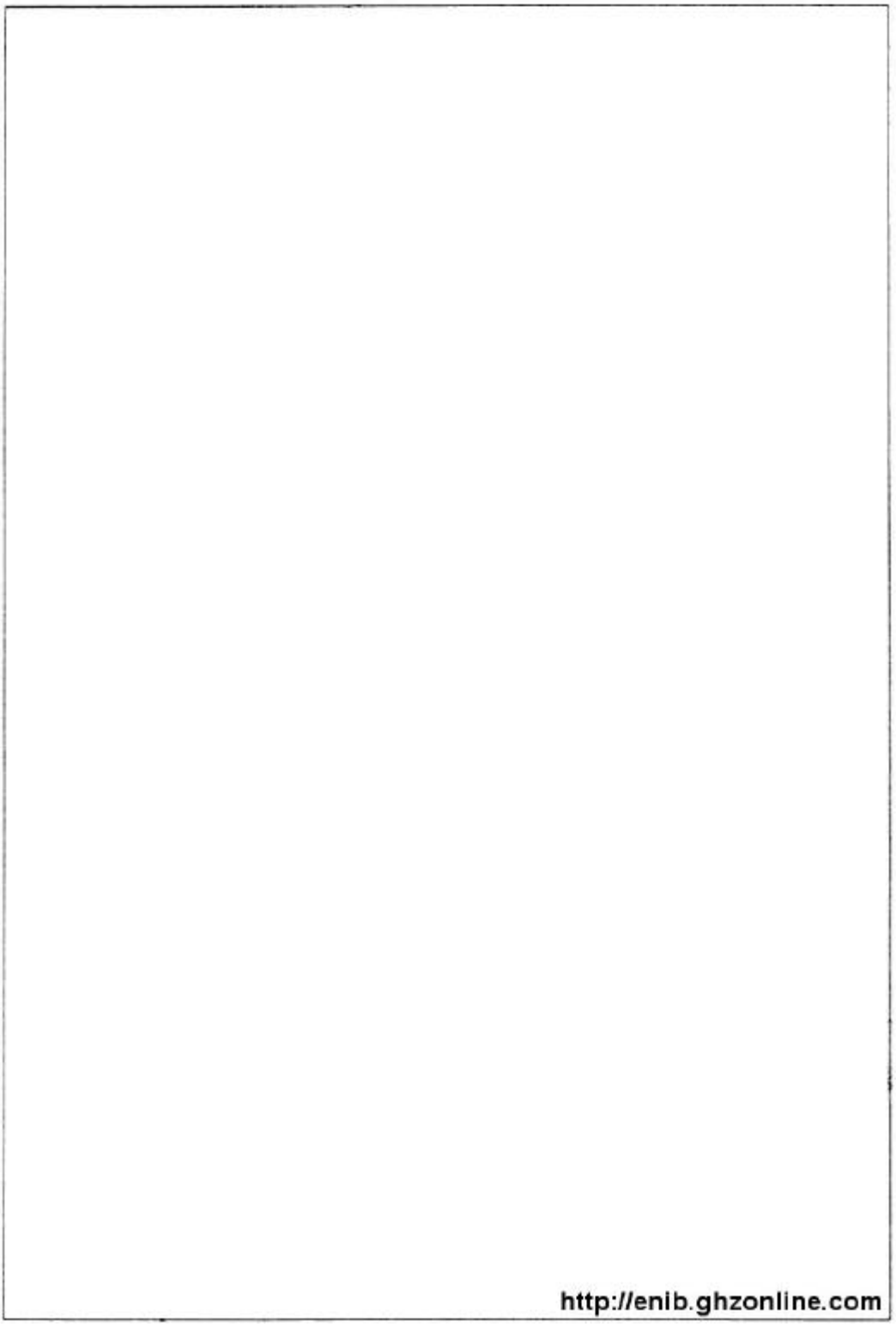
- Le bras d'un shadock est un entier qui vaut
  - 1 si le bras est en haut
  - 0 si le bras est en bas.

- Quand un Shadock pompe à gauche, son bras gauche bouge (il va en bas s'il était en haut et réciproquement).
- Un mécanisme similaire anime le bras droit lorsqu'il pompe droite.
- Lorsque le Shadock pompe à gauche, il vide l'espace de son air (la quantitée expulsée par la pompe diminue de 0.1).
- Lorsque le shadock pompe droite, il remplit l'espace (la quantitée pompée par la pompe est incrémentée de 0.1).
- Un Shadock est bête, aussi, à chaque instant, le choix du sens de pompage est tiré aléatoirement.
- N'oubliez pas que :
  - Pour obtenir un nombre aléatoire compris entre 0 et n-1, il faut utiliser la fonction `int rand()` de cette façon :

```
int monNombre;
monNombre = rand() % n ;
```

  - Pour convertir un ascii en entier, on peut utiliser la fonction `atoi` (voire poly de C) qui se trouve dans `stdlib.h`.

Ecrire la fonction `main` des Shadocks :



3 Annexes

3.1 Fonctions des shadocks

```
/*-------------------------------------------------------------*/
void pompeADroite(Shadock *unShadock, PompeAire *unePompe){
  if(unShadock->brasDroit == 1)
    {
      unShadock->brasDroit=1;
    }
  else
    {
      unShadock->brasDroit=0;
    };
  unePompe->quantitee+=0.1;
 printf("%s pompe  droite \n", unShadock->petitNom);
}
/*-------------------------------------------------------------*/


/*-------------------------------------------------------------*/
void  pompeAGauche(Shadock *unShadock, PompeAire *unePompe){
  if(unShadock->brasGauche == 1)
    {
      unShadock->brasGauche=0;
    }
  else
    {
      unShadock->brasGauche=1;
    }
  unePompe->quantitee-=0.1;
  printf("%s pompe  gauche \n", unShadock->petitNom);
}
/*-------------------------------------------------------------*/


/*-------------------------------------------------------------*/
void afficheUnShadock( Shadock *unShadock){
  printf("Shadock : %s --- bras gauche : %d ---  bras droit :
          %d ---\n", unShadock->petitNom,
          unShadock->brasDroit, unShadock->brasGauche);
}
/*-------------------------------------------------------------*/

/*-------------------------------------------------------------*/
void affichePompeAire(PompeAire *unePompe){
  printf("Pompe : %s ---- quantitee : %f --- \n", unePompe->numSerie,
 unePompe->quantitee);
}
```

```
/*-------------------------------------------------------*/
```

3.2 Exécution du code des Shadocks

Premier exemple : on oublie de passer les arguments en ligne de commande :

```
$ shadock
Erreur :
taper : shadock nomShadock1 nomShadock2 numSeriePompe nbSimulation
```

Deuxime exemple : on lance le programme correctement :

```
shadock "jolly jumper" marsupilami, xTf456erpQ 3

=================================================
Simulation : pas numro 0
jolly jumper pompe  droite
marsupilami, pompe  gauche
Shadock : jolly jumper --- bras gauche : 0 ---  bras droit : 0 ---
Shadock : marsupilami, --- bras gauche : 0 ---  bras droit : 1 ---
Pompe : xTf456erpQ ---- quantitee : 0.000000 ---
=================================================
Simulation : pas numro 1
jolly jumper pompe  droite
marsupilami, pompe  droite
Shadock : jolly jumper --- bras gauche : 0 ---  bras droit : 0 ---
Shadock : marsupilami, --- bras gauche : 1 ---  bras droit : 0 ---
Pompe : xTf456erpQ ---- quantitee : 0.200000 ---
=================================================
Simulation : pas numro 2
jolly jumper pompe  droite
marsupilami, pompe  droite
Shadock : jolly jumper --- bras gauche : 0 ---  bras droit : 0 ---
Shadock : marsupilami, --- bras gauche : 1 ---  bras droit : 1 ---
Pompe : xTf456erpQ ---- quantitee : 0.400000 ---
```