

Programmation Avancée en Langage C

Examen Module I22, Licence Informatique, USTV.

juin 2010

1. **Commande.** Décrire les commandes :

commande	description	exemple
tar	manipulation des archives	tar xvf foo.tar
gcc		
gdb		
make		
indent		
grep		
find		
wc		

2. **Option de compilation.** Donner la signification de la ligne de compilation : `gcc -Wall -g code.o prog.c -o prog.exe`

--

3. **Flux standard.** Préciser les 3 flux standards associés à un processus.

flux	description

Quelle commande permet de rediriger la sortie standard de `prog.exe` vers un fichier `fic.txt` ?

7. Structure.

1. Déclarer un type **complexe** ayant deux champs de type **double**.

2. Ecrire une fonction **double module(complexe z)** pour calculer le module d'un nombre complexe.

3. Ecrire une fonction **void conjugue(complexe *z)** pour calculer sur place le conjugué d'un nombre complexe.

4. Quelle fichier faut-il inclure dans le code source ?

5. Quelle bibliothèque faut-il lier à la compilation ?

8. **Chaine de caractères.** Ecrire une fonction **int sdc(char * str)** qui

calcule et renvoie la somme des chiffres contenus dans la chaîne **str**.

4. Erreur d'exécution.

```
1 int main( void )
2 {
3 int x = 0;
4 int *y = NULL;
5
6 return 0;
7 }
```

Quelle instruction suffirait-il d'insérer en ligne 5 pour obtenir un programme se terminant mal :

instruction	terminaison
	Erreur de segmentation
	Exception en point flottant
	Débordement de pile

5. fonction main.

Décrire le rôle et l'usage d'une fonction **main**.

6. Makefile.

```
CF= -Wall -g
all : prog.exe
prog.exe : prog.c
gcc $(CF) prog.c -o prog.exe
```

```
demo : prog.exe
./prog.exe 123
```

On utilise le fichier **makefile** pour gérer la compilation du programme de **prog.c**.

1. Quelle commande faut-il lancer pour obtenir la compilation du programme **prog.c**.

2. Quelle commande faut-il lancer pour exécuter la démonstration associée à la cible **demo** ?

8. Mémoire.

```
char *x;
int main ( void )
{
char *y = NULL;
x = (char*) malloc(1);
return 0;
}
```

La mémoire associée à un programme est divisée en 4 zones. Préciser le nom et le rôle de ces zones. Dans quelle zone sont situées les adresses correspondant aux expressions : **x**, **&x**, ***x**, **y**, **&y**, ***y**, **&main**, avant le retour de **main**.

zone	description	expression
hors zone		
	n'est pas une adresse	