

Examen du 10 mars 2003 VERSION CORRIGÉE

Tous documents autorisés. Durée 2h.

1 Tests Structurels

Exercice 1

Le programme suivant détermine le signe du produit de ses arguments.

Programme :

```
int signe (int e1, e2)
{
    int s_e1 = 1;
    int s_e2 = -1;
    int res = NEGATIF;

    if ( e1 == 0 || e2 == 0 )
        { res = ZERO; }
    else {
        if ( e1 < 0 )
            { s_e1 = -1; }

        if ( e2 > 0 )
            { s_e2 = 1; }

        if ( s_e1 == s_e2 )
            { res = POSITIF; }
    }

    return res;
}
```

Questions :

1. Déterminer les entrées et sorties du programme ;
2. Dessiner le graphe de contrôle de programme (organigramme) ;
3. Déterminer les jeux de tests pour obtenir la *couverture des instructions* du programme ;
4. Déterminer les jeux de tests pour obtenir la *couverture des chemins* du programme ;
5. Déterminer les jeux de tests pour obtenir la *couverture des conditions* du programme.

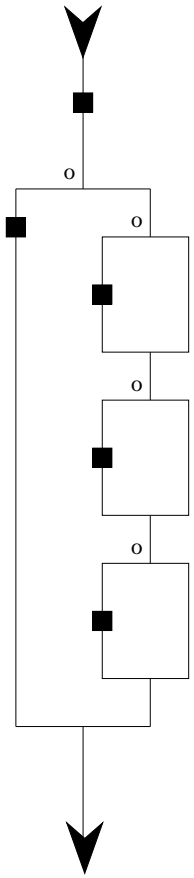
Rappel : Les commentaires associés à chaque test doivent être explicites et faire référence aux parties testées.

Solution :

Entrées : e1, e2

Sorties : signe

Graphe de contrôle :



```

s_e1 = 1
s_2 = -1
res = NEGATIF

e1 == 0 // e1 == 0
res = ZERO
e1 < 0

s_e1 = -1

e2 > 0

s_e2 = 1

s_e1 == s_e2
res = POSITIF

```

Les tests des instructions

Commentaire	e1	e2	signe
e1=0 ou e2=0	0	0	ZERO
e1>0, e2>0	2	3	POSITIF
e1<0, e2>0	-5	4	NEGATIF

Les tests des chemins

Commentaire	e1	e2	signe
e1=0 ou e2=0	0	0	ZERO
e1>0, e2>0	2	3	POSITIF
e1>0, e2<0	5	-4	NEGATIF
e1<0, e2>0	-10	1	NEGATIF
e1<0, e2<0	-6	-7	POSITIF

Les tests des conditions

Commentaire	e1	e2	signe
e1≠0, e2=0	1	0	ZERO
e1=0, e2≠0	0	-2	ZERO
e1=0, e2=0	0	0	ZERO
e1>0, e2>0	3	4	POSITIF
e1>0, e2<0	5	-6	NEGATIF
e1<0, e2>0	-7	8	NEGATIF

Exercice 2

Une horloge radio-contrôlée contacte toutes les heures une horloge atomique pour se mettre à l'heure si elle a dérivé. Si elle n'arrive pas à la contacter, elle retourne le nombre d'heures depuis le dernier contact sur 2 chiffres, plafonné à 99 heures.

Programme :

```
#define MAX_HEURES 99
#define DELTA 2

typedef struct tm {
    int hour; /* 0 - 23 */
    int min; /* 0 - 59 */
};

int h_aff = 0;

int synchro_horloge (tm *temps) {
    struct tm t_at;
    int reception = 0;
    int nb_test = 0;

    if ( temps->min * temps->hour == 0 ) {
        /* Il est temps de se synchroniser */

        reception = acces_horloge_atomique (&t_at);

        /* L'horloge a-t-elle été reçue ? */
        if (reception) {
            /* Mise à jour de l'horloge */
            if ( abs ((temps->hour*60+temps->min) -
                    (t_at.hour*60+t_at.min)) > DELTA ) {
                temps->min = t_at.min;
                temps->hour = t_at.hour;
            }
            h_aff = 0;
        }
        else {
            /* Incrémentation du nombre de tentatives infructueuses */
            if ( h_aff < MAX_HEURES )
                h_aff = h_aff + 1;
            else
                h_aff=MAX_HEURES;
        }
        return h_aff;
    }
}
```

Questions :

1. Déterminer les entrées et sorties du programme ;
2. Dessiner le graphe de contrôle de programme (organigramme) ;
3. Déterminer les jeux de tests pour obtenir la *couverture des données aux limites* du programme ;
4. Déterminer les jeux de tests pour obtenir la *couverture des données hors limites* du programme.

Solution :

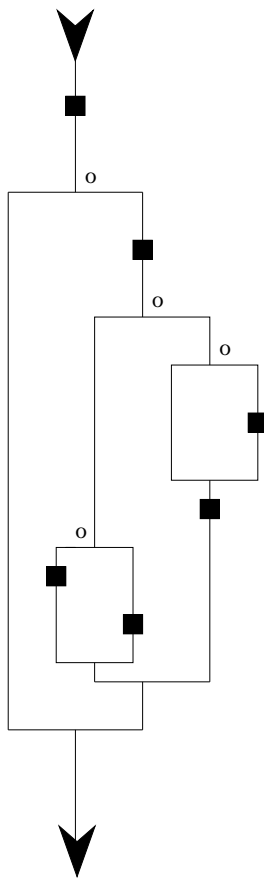
Entrées :

- h_aff : Compteur global d'erreur
- temps->min, temps->hour : valeur de l'horloge
- reception, t_at : sorties de la fonction acces_horloge_atomique

Sorties :

- temps->min, temps->hour : valeur de l'horloge
- h_aff : Compteur global d'erreur

Graphe de contrôle



```
reception = 0
nb_test = 0

temps->min * temps->hour == 0

reception = access_horloge_atomique(*t_at)

reception
abs((temps->hour*60+temps->min)
    (t_at.hour*60+t_at.min)) < DELTA
temps->hour = t_at.hour
temps->min = t_at.min

h_aff = 0
h_aff < MAX_HEURES
h_aff = h_aff + 1
h_aff = MAX_HEURES
```

Valeurs fonctionnelles

- h_aff : 0 .. 99
- temps->min : 0 .. 59
- temps->hour : 0 .. 23
- reception : Booléen
- t_at.min : 0 .. 59
- t_at.hour : 0 .. 23

Valeurs limites informatiques

- h_aff : min_int .. max_int
- temps->min : min_int .. max_int
- temps->hour : min_int .. max_int

- reception : Booléen
- t_at.min : 0 .. 59 par hypothèse
- t_at.hour : 0 .. 23 par hypothèse

Test aux limites

Commentaire	h_aff	temps->hour	temps->min	reception	t_at.hour	t_at.min	h_aff	temps->hour	temps->min
point fonc h_aff	98	10	0	false	-	-	99	10	0
limite basse h_aff	0	9	0	false	-	-	0	9	0
limite haute h_aff	99	9	0	false	-	-	99	9	0
point fonc DELTA > 2	54	15	0	true	15	3	0	15	3
point fonc DELTA > 2	54	15	0	true	14	57	0	14	57
point fonc DELTA <= 2	0	15	0	true	15	2	0	15	0
point fonc DELTA <= 2	32	15	0	true	14	58	0	15	0
point fonc temps->hour et limite basse temps->hour ?	32	0	10	true	0	10	0	0	10
point fonc temps->min et limite basse temps->min	0	7	0	true	7	0	0	7	0
limite haute temps->min et temps->hour	0	23	59	-	-	-	0	23	59
limite haute t_at.min	0	10	0	true	8	59	0	8	59
limite basse t_at.min	0	8	0	true	8	1	0	1	0
limite haute t_at.hour	0	17	0	true	23	0	0	17	0
limite basse t_at.hour	0	19	0	true	0	1	0	0	1

Test hors limites

Commentaire	h_aff	temps->hour	temps->min	reception	t_at.hour	t_at.min	h_aff	temps->hour	temps->min
hors limite haute h_aff	110	10	0	false	-	-	99	10	0
hors limite basse h_aff ?	-1	9	0	false	-	-	0	9	0
hors limite basse h_aff ?	-1	9	0	true	9	0	0	9	0
limite haute temps->min	0	0	62	true	1	33	0	1	33
limite haute temps->min	0	10	62	false	-	-	0	10	62
limite basse temps->min	0	0	-10	true	1	33	0	1	33
limite basse temps->min	0	7	-10	false	-	-	0	7	-10
limite haute temps->hour	0	29	0	true	14	57	0	14	57
limite haute temps->hour	0	29	10	false	-	-	0	29	10
limite basse temps->hour	0	-15	0	true	14	57	0	14	57
limite basse temps->hour	0	-1	16	false	-	-	0	-1	16

Remarque : remise à jour toutes les minutes à minuit.

2 Tests Fonctionnels

Exercice 3

Il s'agit de tester la commande Unix **head** dont voici ci-dessous un extrait du manuel d'utilisation.

```
HEAD(1)                Manuel de l'utilisateur Linux                HEAD(1)
```

NOM

head - Afficher le début d'un fichier.

SYNOPSIS

```
head [-c N[bk]] [-q] [fichier...]  
head [--version]
```

DESCRIPTION

Cette page de manuel documente la version GNU de **head**.

head affiche la première partie (10 lignes par défaut) de chacun des fichiers mentionnés. La lecture se fait sur l'entrée standard si aucun fichier n'est indiqué, ou si le nom '-' est fourni.

Si plusieurs fichiers sont fournis, un en-tête est affiché avant chacun d'eux, mentionnant le nom du fichier entre '==>' et '<=='.
'==>' et '<=='.

OPTIONS

-c N

Afficher les N premiers octets. N est un entier positif, suivi éventuellement d'une lettre indiquant une unité différente de l'octet.

b Bloc de 512 octets.

k Bloc de 1 Ko.

-q

Ne pas afficher les en-têtes mentionnant les noms de fichiers.

--version

Afficher un numéro de version sur la sortie standard et terminer normalement.

Questions :

1. Déterminer les entrées et sorties de la commande ;
2. Déterminer les jeux de tests pour obtenir la *couverture nominale* de la commande.
3. Déterminer les jeux de tests pour obtenir la *couverture aux et hors limites* de la commande.

Solution :

Entrées :

- options : --version, -c Nbk, -q
- paramètres : fichiers

Sorties :

- écran

Tests nominaux

Commentaire	option	paramètre	Affichage à l'écran
option version	--version		La version
sans option		fichier	Les 10 premières lignes du fichier
sans option			Les 10 premières lignes de l'entrée standard
sans option		-	Les 10 premières lignes de l'entrée standard
option -c	-c 15	fichier	Les 15 premiers octets du fichier
option -c	-c 8b	fichier	Les 8 premiers blocs du fichier
option -c	-c 17k	fichier	Les 17 premiers kilo-octets du fichier
option -q	-q	fichier	Les 10 premières lignes du fichier
option -q	-q	fic1 fic2 fic3	Les 10 premières lignes des fichiers fic1 à fic3 sans entête
plusieurs fichiers		fic1 fic2 fic3	Affichage des 10 premières lignes des fichiers fic1 à fic3
plusieurs fichiers	-c 500b	fic1 fic2 fic3 fic4	Les 500 premiers blocs des fichiers fic1 à fic4
plusieurs fichiers	-c 15	fic1 fic2 - fic4	Les 15 premières lignes des fichiers fic1 à fic2, entrée standard puis fic4

Tests aux limites

Commentaire	option	paramètre	Affichage à l'écran
option	-c 1	fichier	La premier ligne du fichier
option	-c 1b	fichier	Le premier bloc du fichier
option	-c 1k	fichier	Le premier kilo-octet du fichier
sans option		f1 .. fn	Les 10 premières lignes des fichiers f1 .. fn

Tests hors limites

Commentaire	option	paramètre	Affichage à l'écran
erreur syntaxe	11		Erreur
option erronée	-z		Erreur
option erronée	--errone		Erreur
option -c négatif	-c -10	-	Erreur
option -c nul	-c 0	-	Erreur
fichier inexistant		fic_erroné	Erreur