Université Paris 6 - CNAM
DESS DLS

Module de test du logiciel
P. Ayrault - V. Ménissier-Morain

Examen du 27 février 2001

Tous documents autorisés. Durée 2h.

1 Tests Structurels

Exercice 1

Nous avons un programme qui détermine le gagnant d'un jeu de cartes. Les règles du jeu sont les suivantes :

- les cartes par ordre croissant vont du 2 au 10 puis Valet, Dame, Roi et As (les couleurs ne comptent pas)
- chaque joueur tire 2 cartes
- le joueur ayant la paire la plus forte gagne
- si aucun des joueurs n'a de paire, celui qui a la carte la plus forte gagne.

Programme

```
void Determine_Gagnant(CA1, CA2, CB1, CB2, Gagnant);
carte_T CA1, CA2;         /* cartes du joueur A */
carte_T CB1, CB2;         /* cartes du joueur B */
gagnant_T * Gagnant;      /* Nom du gagnant */
{
  bool Paire_A;
  bool Paire_B = False;

  /* Determiner si joueur A a une paire */
  if ( CA1 == CA2 )                                 /* C1 */
    Paire_A = True;
  else
    Paire_A = False;

  /* Determiner si joueur B a une paire */
  if ( CB1 == CB2)                                  /* C2 */
    Paire_B = True;

  /* Le gagnant a la meilleure des paires */
  if ( Paire_A ) {                                  /* C3 */
    if ( Paire_B ) {                                /* C4 */
      if ( CA1 > CB1 )                              /* C5 */
        *Gagnant = Joueur_A;
      else
        if ( CA1 < CB1 )                            /* C6 */
          *Gagnant = Joueur_B;
        else
          *Gagnant = Egalite;
    }
```

```
        else
          *Gagnant = Joueur_A;
      }
      else
        if ( Paire_B )                                    /* C7 */
          *Gagnant = Joueur_B;
        else {
          /* Aucun des joueurs n'a de paire */
          Meilleure_A = CA2;
          Meilleure_B = CB2;

          if ( CA1 > CA2 )                                /* C8 */
            Meilleure_A = CA1;

          if ( CB1 > CB2 )                                /* C9 */
            Meilleure_B = CB1;

          if ( Meilleure_A > Meilleure_B )                /* C10 */
            *Gagnant = Joueur_A;
          else
            if ( Meilleure_B > Meilleure_A )              /* C11 */
              *Gagnant = Joueur_B;
            else
              *Gagnant = Egalite;
        }
    }
```

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.

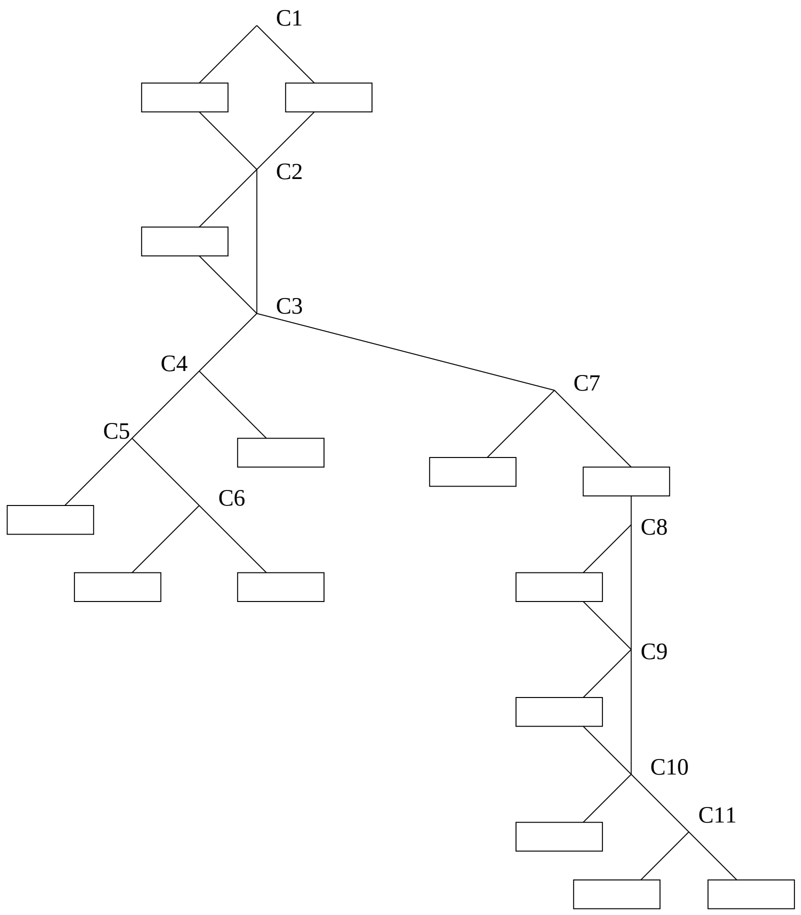
Rappel : Les commentaires associés à chaque test doivent être explicites et faire référence aux parties testées.

Solution :

Entrées : CA1, CA2, CB1, CB2

Sorties : Gagnant

Graphe de contrôle :



Les tests des instructions :

Commentaire	**CA1**	**CA2**	**CB1**	**CB2**	**Gagnant**
A a la meilleure paire (C5)	10	10	3	3	Gagnant_A
B a la meilleure paire (C6)	9	9	V	V	Gagnant_B
A et B même paire (!C6)	8	8	8	8	Egalite
paire A et non paire B (!C4)	4	4	V	D	Gagnant_A
paire B et non paire A (C7)	3	2	7	7	Gagnant_B
non paire A et B, A a la meilleure carte (C10)	10	4	8	5	Gagnant_A
non paire A et B, B a la meilleure carte (C11)	7	6	R	3	Gagnant_B
non paire A et B, A et B ont la même meilleure carte (!C11)	As	4	As	8	Egalite

Les tests des chemins :

Commentaire	CA1	CA2	CB1	CB2	Gagnant
A a la meilleure paire (C1, !C2,C3,C4,C5)	10	10	3	3	Gagnant_A
B a la meilleure paire (C1, !C2,C3,C4, !C5,C6)	9	9	V	V	Gagnant_B
A et B même paire (C1, !C2,C3,C4, !C5, !C6)	8	8	8	8	Egalite
paire A et non paire B (C1,C2,C3, !C4)	4	4	V	D	Gagnant_A
paire B et non paire A (!C1, !C2, !C3,C7)	3	2	7	7	Gagnant_B
non paire A et B, A a la meilleure 1ère carte (!C1, !C2, !C3, !C7,C8,C9,C10)	10	4	8	5	Gagnant_A
non paire A et B, A a la meilleure 1ère carte (!C1, !C2, !C3, !C7,C8, !C9,C10)	10	4	5	8	Gagnant_A
non paire A et B, A a la meilleure 2de carte (!C1, !C2, !C3, !C7, !C8,C9,C10)	4	10	8	5	Gagnant_A
non paire A et B, A a la meilleure 2de carte (!C1, !C2, !C3, !C7, !C8, !C9,C10)	4	10	5	8	Gagnant_A
non paire A et B, B a la meilleure 1ère carte (!C1, !C2, !C3, !C7,C8,C9, !C10,C11)	7	6	R	3	Gagnant_B
non paire A et B, B a la meilleure 1ère carte (!C1, !C2, !C3, !C7, !C8,C9, !C10,C11)	6	7	R	3	Gagnant_B
non paire A et B, B a la meilleure 2de carte (!C1, !C2, !C3, !C7,C8, !C9, !C10,C11)	7	6	3	R	Gagnant_B
non paire A et B, B a la meilleure 2de carte (!C1, !C2, !C3, !C7, !C8, !C9, !C10,C11)	6	7	3	R	Gagnant_B
non paire A et B, A et B ont la même meilleure carte (!C1, !C2, !C3, !C7,C8,C9, !C10, !C11)	As	4	As	8	Egalite
non paire A et B, A et B ont la même meilleure carte (!C1, !C2, !C3, !C7, !C8,C9, !C10, !C11)	4	As	As	8	Egalite
non paire A et B, A et B ont la même meilleure carte (!C1, !C2, !C3, !C7,C8, !C9, !C10, !C11)	4	As	8	As	Egalite
non paire A et B, A et B ont la même meilleure carte (!C1, !C2, !C3, !C7, !C8, !C9, !C10, !C11)	4	As	8	As	Egalite

Exercice 2

Nous avons un programme qui calcule la validité d'une tension lue, suivant les exigences suivantes :

- si la tension lue est hors des bornes fonctionnelles $[0..TENSION\_MAX]$, le capteur est déclaré invalide
- lorsque la tension est entre ses bornes fonctionnelles, si

$$|\texttt{tension lue} - \texttt{tension cycle précédent}| > DELTA\_MAX$$

  pendant 2 cycles consécutifs, alors le capteur n'est plus valide. Pendant ce filtrage, la tension filtrée reste à la valeur précédente et le capteur reste valide
- dans tous les autres cas, le capteur est déclaré valide et la tension filtrée est égale à la tension lue.

Programme

```
#define TENSION_MAX     380;
#define DELTA_MAX        40;
#define FILTRE_MAX        2;
```

```
int_16 tension_pre;
int_16 filtre = 0;

void filtrage_entrees (tension_lue, tension_filtree, validite_capteur);
int_16 tension_lue;
int_16 *tension_filtree;
bool *validite_capteur;
{
  /* Verification de la gamme de la tension lue */
  if (( tension_lue < 0 ) || (tension_lue > TENSION_MAX))   /* C1 */
    /* capteur HS */
    *validite_capteur = False;
  else
    /* Coherence tension lue */
    if (( tension_lue - tension_pre <= DELTA_MAX) ||         /* C2 */
       ( tension_pre - tension_lue >= DELTA_MAX)) {
       /* Tension lue incoherente */
      if ( filtre >= FILTRE_MAX ) {                          /* C3 */
        /* Filtrage maximum atteint */
        *validite_capteur = False;
      }
      else {
        *validite_capteur = True;
        *tension_filtree = tension_pre;
        /* Increment du filtrage */
        filtre = filtre + 1;
      }
    }
    else {
      /* Initialisation des sorties */
      tension_filtree = tension_lue;
      *validite_capteur = True;
      *tension_pre = tension_lue

      /* Reset du filtrage */
      filtre = 0;
    }
}
```

Questions :

1. Déterminer les entrées et sorties du programme ;
2. Déterminer les jeux de tests pour obtenir la *couverture des données aux limites* du programme ;
3. Déterminer les jeux de tests pour obtenir la *couverture des données hors limites* du programme.

Solution :

Entrées : tension_lue, filtre, tension_pre
Sorties : tension_filtree, validite_capteur, tension_pre, filtre

Valeurs fonctionnelles de la tension : $0..TENSION\_MAX$

Commentaire	tension_lue	filtre	tension_pre	tension_filtree	validite_capteur	tension_pre	filtre
limite basse fonctionnelle	-1	0	220	?	False	220	0
limite basse fonctionnelle	0	0	10	0	True	0	0
limite haute fonctionnelle	381	0	220	?	False	220	0
limite haute fonctionnelle	380	0	360	380	True	380	0
hors limite delta tension	220	0	261	261	True	261	1
hors limite delta tension	220	1	179	179	True	179	2
dans limite delta tension	220	1	260	220	True	220	0
dans limite delta tension	220	1	180	220	True	220	0
hors limite filtrage	220	2	170	220	False	?	2

Valeurs limites informatiques : $-Max\_int16..Max\_int16$.

Commentaire	tension_lue	filtre	tension_pre	tension_filtree	validite_capteur	tension_pre	filtre
limite basse informatique	-Max_int	0	170	220	False	?	0
limite haute informatique	Max_int	0	170	220	False	?	0

2 Tests Fonctionnels

Exercice 3

Il s'agit de tester la commande Unix **cal** dont le manuel d'utilisation est décrit ci-dessous.

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


NOM
       cal - Afficher un calendrier.

SYNOPSIS
       cal [-mjy] [[mois] [annee]]

DESCRIPTION
       Cal affiche un calendrier.  Si les arguments sont omis, le
       mois courant  est  affiché  sauf  pour  l'option  -y.  Les
       options sont les suivantes:

       -m     Affiche le calendrier en considérant lundi comme le
       premier jour de la semaine (Dimanche par défaut.)

       -j     Affiche  les  numéros  des  jours  dans l'année (le
              numéro 1 correspond au 1er Janvier).

       -y     Affiche le calendrier de l'année courante.

       Un paramètre unique indique l'année (1 - 9999) à afficher;
```

```
noter que l'année doit être complètement spécifiée : ``cal
89'' n'affiche pas le calendrier de 1989.  Deux paramètres
indiquent  le mois (1 - 12) et l'année.  S'il n'y a pas de
paramètres, le calendrier du mois courant est affiché.

L'année commence le 1er Janvier.
```

HISTORIQUE

La commande **cal** est apparue dans la version 6 de AT&T UNIX.

TRADUCTION

Michel Quercia, 1997.

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 limites* de la commande lorsqu'il n'y a pas d'option ;
4. Déterminer les jeux de tests pour obtenir la *couverture hors limites* de la commande lorsqu'il n'y a pas d'option.

Solution :

Entrées :

– options : -j, -m, -y
– paramètres : jour, année

Sorties : Calendrier

Tests nominaux :

Paramètre	Option			Résultats
	-m	-j	-y	
Sans paramètre				Affichage mois courant commençant par le dimanche
Sans paramètre	x			Affichage mois courant commençant par lundi
Sans paramètre		x		Affichage numéro jour du mois courant
Sans paramètre			x	Affichage année courante
Année				Affichage de l'année demande commençant par le dimanche
Année	x			Affichage de l'année demande commençant par le lundi
Année		x		Affichage numéro des jours de l'année
Année			x	Message d'erreur
Mois, année				Affichage du mois de l'année demande commençant par le dimanche
Mois, année	x			Affichage de l'année demande commençant par le lundi
Mois, année		x		Affichage numéro des jours de l'année
Mois, année			x	Message d'erreur

Tests aux limites sans option :

Paramètre	Option			Résultats
	-m	-j	-y	
1				Affichage de l'année 1
9999				Affichage de l'année 9999
1, 1				Affichage du mois de janvier de l'an 1
12, 9999				Affichage du mois de décembre de l'an 9999

Tests hors limites sans option :

Paramètre	Option			Résultats
	-m	-j	-y	
0				?
10000				?
0, 2001				?
13, 2001				?
-1				?
a				?
. . .				?