



POLYTECHNIQUE
MONTRÉAL

Questionnaire Contrôle Périodique 1

LOG3430

Sigle du cours

Identification de l'étudiant(e)		
Nom :	Prénom :	
Signature :	Matricule :	Groupe :

Sigle et titre du cours		Groupe	Trimestre
LOG3430 - Méthodes de test et de validation du logiciel		Tous	20151
Professeur		Local	Téléphone
Giuliano Antoniol		C-624	
Jour	Date	Durée	Heures
Vendredi	20 février 2015	1 heure	
Documentation		Calculatrice	
<input type="checkbox"/> Aucune <input checked="" type="checkbox"/> Toute <input checked="" type="checkbox"/> Voir directives particulières		<input type="checkbox"/> Aucune <input checked="" type="checkbox"/> Toutes <input type="checkbox"/> Non programmable	Les connections aux réseaux sont interdits.
Directives particulières			
Toute documentation est permise, ainsi que les calculatrices, à l'exception toutefois des de tout dispositif connectés a Internet.			

Important	Cet examen contient <input type="text" value="2"/> exercice et <input type="text" value="0"/> question sur un total de <input type="text" value="6"/> pages (excluant cette page)
	La pondération de cet examen est de <input type="text" value="5"/> %
	Vous devez répondre sur : <input checked="" type="checkbox"/> le questionnaire <input type="checkbox"/> le cahier <input type="checkbox"/> les deux
	Vous devez remettre le questionnaire : <input checked="" type="checkbox"/> oui <input type="checkbox"/> non

L'étudiant doit honorer l'engagement pris lors de la signature du code de conduite.

Exercice 1 – 20 points

Considérez le programme suivant :

```
int min2int(int x,int y){ /* 1 */
    if (x>y) /* 2 */
        return x; /* 3 */
    return y; /* 4 */
}

static int distance (const char * w1, int len1, const char * w2, int len2) /* 5
    */
{
    int dist[len1 + 1][len2 + 1], i,j,c1,c2, min;

    for (i = 0; i <= len1; i++) dist[i][0] = i; /* 6 */

    for (i = 0; i <= len2; i++) dist[0][i] = i; /* 7 */

    for (i = 1; i <= len1; i++) { /* 8 */

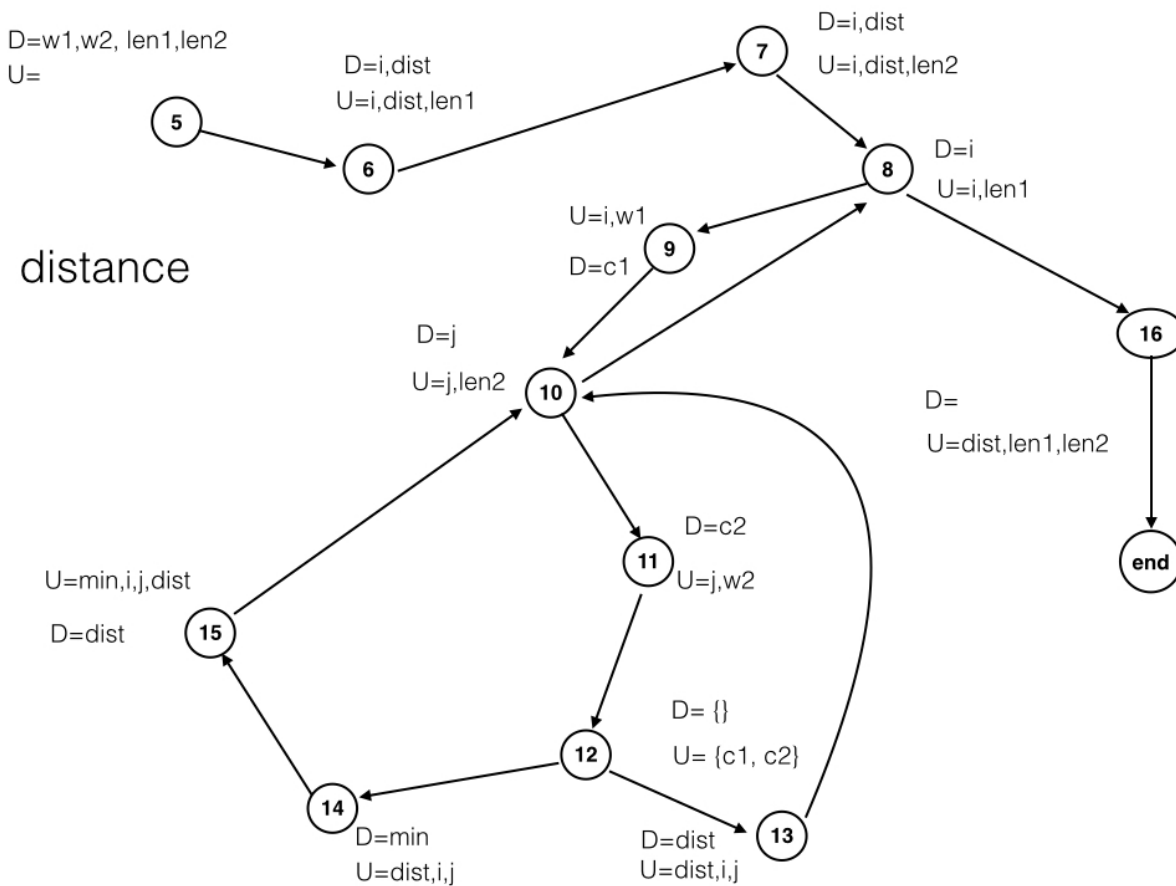
        c1 = w1[i-1]; /* 9 */
        for (j = 1; j <= len2; j++) { /* 10 */
            c2 = w2[j-1]; /* 11 */
            if (c1 == c2) { /* 12 */
                dist[i][j] = dist[i-1][j-1]; /* 13 */
            }else {

                min = min2int (dist[i-1][j], dist[i][j-1])+1; /* 14 */
                dist[i][j] = min2int (min, dist[i-1][j-1] + 2); /* 15 */

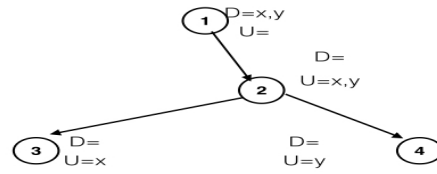
            }
        }
    }
    return dist[len1][len2]; /* 16 */
}
```

- 1) Complétez le *Graphe de flux de contrôle* suivant, où les numéros de nœuds correspondent aux numéros de lignes. Indiquez :
- Les arcs du graphe. (2 points)
 - Les ensembles des définitions et utilisations pour chaque nœud; voir l'exemple pour nœud 12 qui utilise les variables 'c1, c2' et dont l'ensemble des définitions est vide. (2 points)

Réponse à l'exercice 1.1 :



min2int



- 2) Complétez le tableau suivant en donnant les c-uses et les p-uses pour chaque variable. Voir l'exemple pour la variable 'x' qui a une p-use à la ligne x (2 points)

	Variable			
	x		y	
	c-use	p-use	c-use	p-use
1				
2		x		x
3	x			
4			x	

	Variable																			
	i		j		len1		len2		w1		w2		c1		c2		dist		min	
	c-use	p-use	c-use	p-use	c-use	p-use	c-use	p-use	c-use	p-use	c-use	p-use	c-use	p-use	c-use	p-use	c-use	p-use	c-use	p-use
5																				
6	x	x				x											x			
7	x	x						x									x			
8	x	x				x														
9	x								x											
10			x	x				x												
11			x								x									
12	x		x											x		x				
13																	x			
14	x		x														x			
15	x		x														x		x	
16																	x			

- 3) Complétez le tableau suivant en donnant toutes les définitions-utilisations (def-uses) des données. Voir l'exemple pour la variable 'x' dont la définition à la ligne 1 est utilisée à la ligne 2. Si la définition d'une variable n'est pas utilisée, indiquez ceci à l'aide de l'ensemble vide comme montré pour la définition de la variable 'x' à la ligne 4. (4 points)

	Ligne				définition
Variable	1	2	3	4	
x	2,3				
y	2,4				{}

	Ligne définition											
	5	6	7	8	9	10	11	12	13	14	15	16
i		6	7	8,9,13,14,15								
j						10,11,13,14,15						
len1	6,8,16											
len2	7,10,16											
w1	9											
w2	11											
c1					12							
c2							12					
dist		6,7,13,14 15,16	7,13,14 15,16						13,14, 15,16		13,14, 15,16	
min										15		

- 4) Complétez le tableau suivant en donnant des valeurs d'entrée pour couvrir le critère all-uses. Précisez les def-uses couvertes pour chaque valeur d'entrée. Voir l'exemple du cas de test T1. Compléter T1 si nécessaire. Longer chaînes des caractères maximal 3 ! (6 points)

The function min2int contains an error is should be $x < y$ to compute the min of 2 integers; the du covered reported in the tables are for the correct code thus for the test $x < y$ and NOT for $x > y$

The most difficult part is to test the dependencies between lines 13,14, 15; to cover def use between 13 and 14(15) there must be two identical characters followed by two different such as un aa ab. To do the contrary we need bc ac (two different and one equal). To cover 13 to 13 we need something like aa aa.

The remaining could be covered with strings such as ad cb that force the loop between 14(15) and itself.

In a similar was one may use a string of 3 characters like acb and bcd. The first char force the execution of 13 the second char guide the loop via line 14. The third different char closes the loop 13 to 14 and 14 to itself. In this case we miss the dependency 13 to 13 so we need a case like aa aa.

	w1	len1	w2	len2	x	y
TC					//	//
TC					<1,2> <1,3>	<1,2> <1,4>

	w1	len1	w2	len2	i	j	c1	c2	min	dist
TC1	A <5,9>	1 <5,16> <5,6> <5,8>	A <5,11>	1 <5,10> <5,16> <5,6>	<6,6> <7,7> <8,13> <8,8> <8,9>	<10,10> <10,11> <10,13>	<9,12>	<11,12>		<13,16> <6,13> <6,16> <6,6> <7,13> <7,16> <7,7>
TC2	acb	3	bca	3	<8,14> <8,15>	<10,14> <10,15>			<14,15>	<13,14> <13,15> <15,13> <15,14> <15,15> <15,15> <15,16> <6,14> <6,15> <7,14> <7,15>
TC3	Aa	2	Aa	2						<13,13>

- 5) Selon vous est ce que le programme fait le bon calcul? C-a-d la distance minimal entra deux chaines de caractères? (2 points)

Justification: No the test in the function `min2int` is reverse it should check `x<y` and not `x>y`; it just suffice consider the sematic of the function name and the line 15, clearly it return the max between `min` and the `dist_2` and thus it makes no sense.

- 6) Est-ce que les cas de tests nécessaires pour la couverture du critère all-definitions seront suffisants pour la couverture du critère couverture des chemins. (1 point)

- 1.1.1 Oui ☐
 1.1.2 Non X
 1.1.3 Oui dans cet exercice mais non en général ☐
 1.1.4 Non dans cet exercice mais oui en général ☐

Justification: all def just require one use to be covered not even all uses thus ot is impossible to cover all paths.

- 7) Est-ce que les cas de tests nécessaires pour la couverture du critère tous les c-use some p-use seront suffisants pour la couverture du critère couverture des chemins. (1 point)

- 1.1.5 Oui ☐
 1.1.6 Non X
 1.1.7 Oui dans cet exercice mais non en général ☐
 1.1.8 Non dans cet exercice mais oui en général ☐

Justification: the criteria just ask for cover all computational uses one time. Consider the matrix of size `N`; `N` is an integer thus there are as many path as integers thus an infinite number of paths.