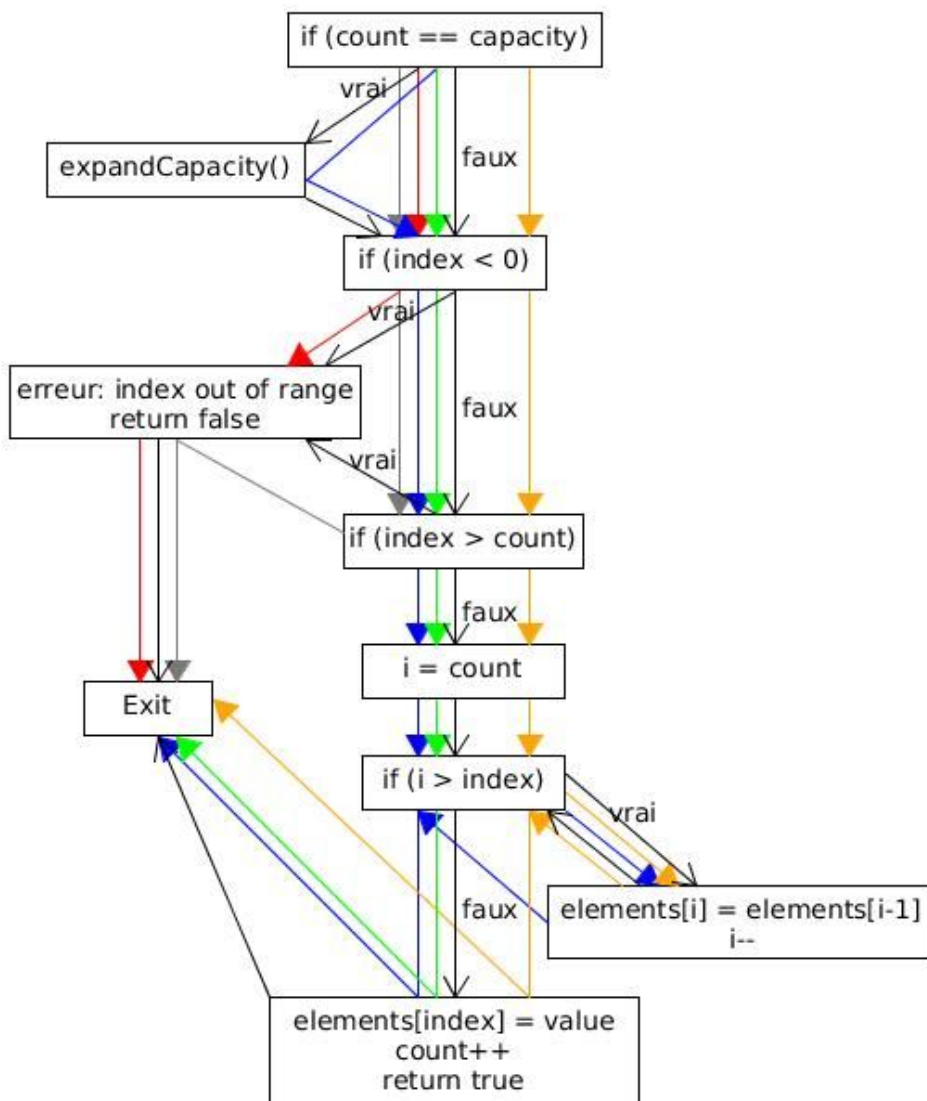


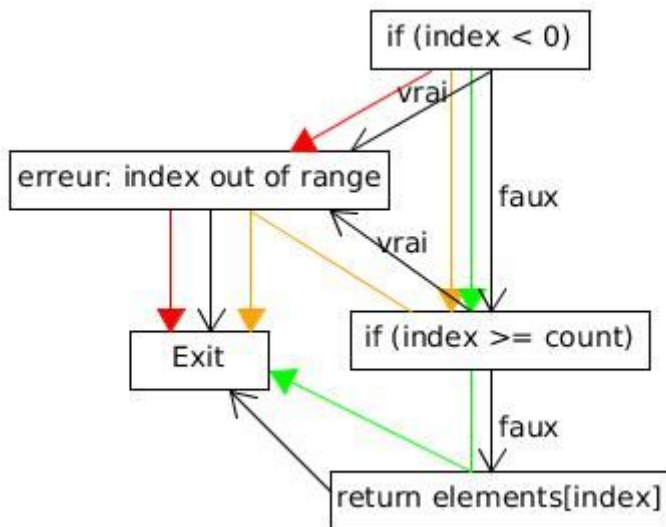
E1) Diagramme de flot de contrôle

1.1

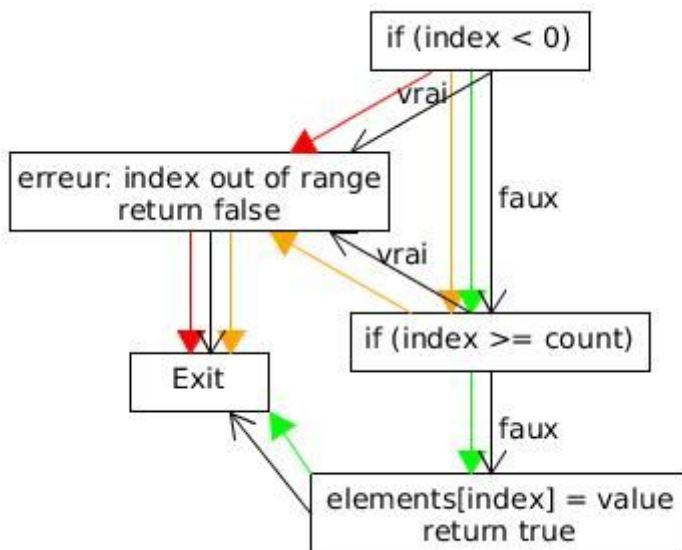
Vector::insert(int index, std::string value);



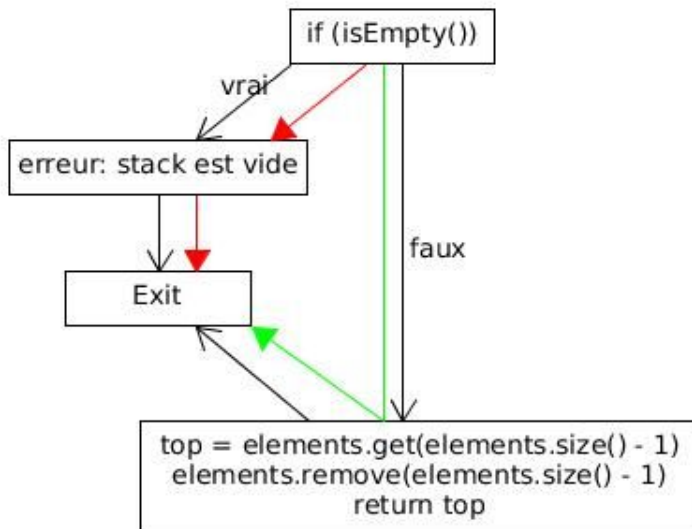
Vector::get(int index);



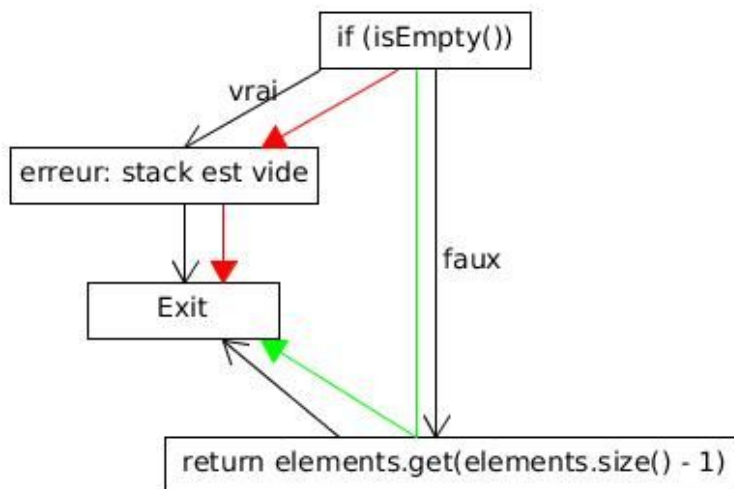
Vector::set(int index, const std::string &value);



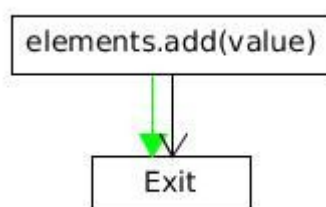
Stack::pop();



Stack::top();



Stack::push(std::string value);



1.2

Fonction	Complexité cyclomatique
Vector::insert()	5
Vector::get()	3
Vector::set()	3
Stack:push()	1
Stack:pop()	2
Stack:top()	2

1.3

Voir Screenshots

E2) Case de tests et jeu de données

Vector::insert()				
Entrée(value)	Entrée(index)	count préalable	capacity préalable	Sortie
"Hello"	12	20	20	true
"Hello"	12	21	40	true
"Hello"	-1	22	40	false
"Hello"	30	22	40	false
Le chemin vert n'est pas faisable car on a préalablement vérifié que (index < count) donc (i (=count) > index) n'est pas possible				

Vector::get()	
avec count = 20 ("Hello" * 20)	
Entrées (index)	Sortie
-1	error
12	"Hello"
50	error

Vector::set()		
avec count = 20		
Entrées (index)	Entrées (value)	Sortie
-1	"Hello"	false
12	"Hello"	true
50	"Hello"	false

Stack::push()	
Entrée(value)	Sortie
"Hello"	<Rien>

Stack::pop()	
Stack préalable	Sortie
OK (top = "Hello")	"Hello"
Empty	error

Stack::top()	
Entrée	Sortie
OK (top = "Hello")	"Hello"
Empty	error

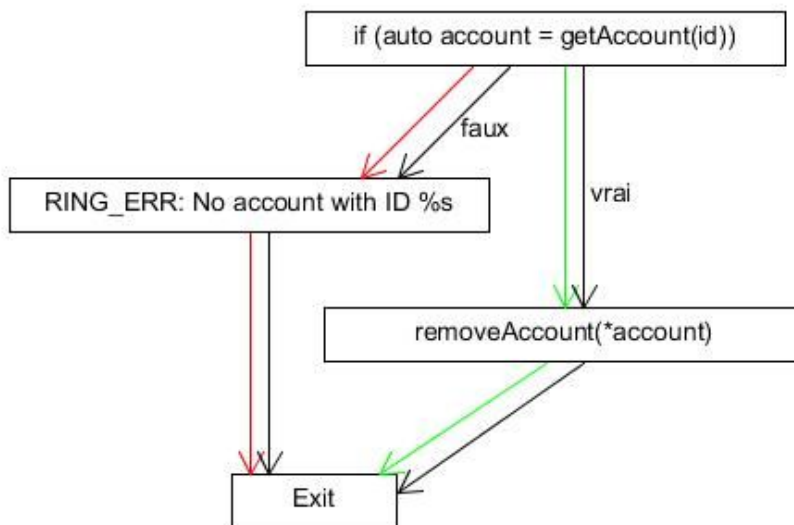
E3) Implémentation et exécution des tests unitaires

Implémentation faite. Incapable d'exécuter correctement.

E4) Contribution au projet Ring

4.1 & 4.3

AccountFactory::removeAccount(const std::string& id);



4.2

Fonction	Complexité cyclomatique
AccountFactory::removeAccount(const std::string& id)	2

4.4

En examinant les fichiers tests dans le dossier test, nous sommes incapables de trouver des tests pour la fonction removeAccount. Ceci dit, les deux chemins possibles n'ont pas été testés.