

SCIPER	Architecture (max3pts)	architecture violation comment	CLASS max(3pts)	CLASS violation comment
283389	2,00	[A1] tu fait plus que necessaire dans le main , pas besoin d afficher le contenu de la ville	3,00	ok
287006	2,00	[A1] projet.cc includes noeud.h	3,00	Ok
288423	3,00	Architecture très bien respectée!	3,00	Interfaces claires et propres avec des fonctions bien externalisées.
288781	2,00	[A1] projet.cc includes error.h et constante.h	0,00	[C1] 14 projet.cc: declaration of an instance of class Ville in the main, it breaks the encapsulation/ 11,12 ville.h; 19 noeud.h: attributes in public. [C2] noeud.h: class Noeud interface is not externalized.
291348	3,00	Ok	0,00	[C2] ville.h, noeud.h, tool.h: only function prototypes should be in .h; allowed exceptions are only getters and constructors that fit onto the same line as the function prototype
295752	2,00	[A1] Class projet.cc includes tools.h and noeud.h, but should only include ville.h as access to the Ville module. Otherwise architecture is correct.	3,00	Good class encapsulation.

295798	2,00	[A1] main.cc calls error::success() ; argc, argv should be treated by main.cc, not by city	3,00	Ok
295810	2,00	Bon respect de l'architecture en général par contre : [A3]tools.cc:17,40 : Le module tools fait appel aux fonctions d'erreurs reliées au projet. Ce module ne doit pas avoir de notion de noeud ou d'erreurs de superposition de noeud et devrait pouvoir être réutilisé pour d'autres applications indépendantes.	2,00	La classe de noeuds est définie correctement [C1]noeud.cc:138,139 : Warning : Il est préférable de définir une classe pour représenter la ville avec un attribut contenant tous les noeuds au lieu de déclarer des variables statiques. [C2]noeud.h:42,52,61 : Les constructeurs sur plusieurs lignes doivent être définis dans noeud.cc.
296202	3,00	OK	1,00	[C2] 31-38 Noeud.h, 23-30 Ville.h : for both the constructor definition is not externalized and doesn't hold on one line.
296232	3,00	[A1] projet.cc includes noeud.h and error.h	3,00	Ok
296253	2,00	[A1] projet.cc includes nodes.h & tools.h	3,00	ok
296351	2,00	[A1] l'instance de ville ne doit pas être déclarée dans projet.cc mais dans le module ville; évitez d'inclure des .h dans d'autres.h (seulement dans les .cc). le module tools est indépendant et ne doit pas inclure error et constantes, pas de pénalité car non utilisés	3,00	encapsulation respectée et interfaces externalisées, bien joué!
296443	1,00	[A1] projet.cc contains an instance of Town, it should only call a function from the Town Module. [A3] tools.h is not completely independant of the project since it contains constantes.h, file specific to this project.	3,00	Care since your getters return the actual lists and not constant references which is the same as setting your list as public (illegal in our guidelines). You do the same for returning structures which contain public variables. Otherwise good job!
296642	3,00	Votre architecture respecte la donnée.	3,00	Bonne encapsulation. Attention néanmoins au getter qui retourne un pointeur sur un noeud dans votre classe ville. Il pourra être susceptible de la briser. Ok car vous ne le modifiez pas mais cela pourrait vous pénaliser si vous ne le rendez pas constant.
296820	3,00	Ok	0,00	[C2] ville.h, noeud.h, tool.h: only function prototypes should be in .h; allowed exceptions are only getters and constructors that fit onto the same line as the function prototype

296826	3,00	Très bonne architecture, bravo !	3,00	Vos classes sont claires et on y trouve facilement le fonctionnement à partir du .h. Bonne utilisation du namespace dans l'ensemble de votre projet.
296831	3,00	OK	2,00	la classe lien hérite d'une struc donc les attributs hérités sont publics, ce qui est interdit pour une classe selon nos conventions
296853	3,00	l'architecture respecte la donnée	3,00	encapsulation respectée et interfaces externalisées, bien joué!
296887	2,00	[A2] class ville is in noeud.h	3,00	Ok
296932	3,00	Ok	3,00	Ok
297104	3,00	Très bonne architecture, bravo !	3,00	Vos classes sont claires et on y trouve facilement le fonctionnement à partir du .h. Bonne utilisation du namespace dans l'ensemble de votre projet.
297186	2,00	[A1] projet includes error.h ; le module tools ne doit pas include constantes.h mais pas de pénalité car non utilisé	1,00	[C1] noeud.cc L20,21 il est strictement interdit d'avoir des attributs publics, à corriger
297202	3,00	Architecture respectée, bravo !	3,00	L'implémentation et l'utilisation de classes dans votre projet est correcte mais les lignes avec le nom de la classe sont inutiles dans votre cas (noeud.h:7-8, par exemple). Vous pourriez aussi mettre la fonction lecture de ville dans la classe Ville.
297280	2,00	Bonne hierchachie des modules mise à part : [A1]projet.cc:29 : l'instance de ville doit être déclarée dans une fonction du module ville, dans votre projet, elle est déclarée dans la fonction main. Définissez une fonction du module ville qui se charge d'instancier la ville et ensuite appeler les méthodes de la classe.	3,00	Bravo très bonne encapsulation et bonne utilisation de l'héritage
297284	3,00	Warning: tools.cc 20 le module tools ne doit pas connaître le concept de noeud. De ce fait, ce mot ne devrait pas se trouver dans le nom de vos fonctions. Cela pourrait vous pénaliser pour les prochains rendus.	2,00	[C2] noeud.h 29,32 les définitions des getters et de la liste d'initialisation du constructeur sont uniquement acceptées si elles tiennent sur une seule ligne. Warning: [C1] noeud.h 28 Vous donnez directement un accès à un attribut privé en retournant un pointeur. Ok car vous n'utilisez pas la fonction pour le modifier mais veuillez à corriger ça.
297293	2,00	[A2] class ville is in noeud.h	3,00	Ok

297301	3,00	Warning : vous vous en êtes bien rendus compte, mais en effet votre classe Ville actuellement ne sert à rien. Elle sera nécessaire dans le futur, le vecteur de Noeud devra en être un attribut, et certaine fonction de détection d'erreur devront donc nécessairement être évaluées dans Ville	3,00	Ok
297768	2,00	Bonne hierchachie des modules mise à part : [A1]projet.cc:29 : l'instance de ville doit être déclarée dans une fonction du module ville, dans votre projet, elle est déclarée dans la fonction main. Définissez une fonction du module ville qui se charge d'instancier la ville et ensuite appeller les méthodes de la classe.	3,00	Bravo très bonne encapsulation et bonne utilisation de l'héritage
297781	3,00	Vous respectez la logique des architectures proposées mais vous rajoutez vos propres fichiers/modules. Evitez cela, surtout qu'ils ne sont pas nécessaire et leurs contenus pourraient être mis dans d'autres fichiers.	3,00	Très bonne classes, bien compréhensibles !
298006	3,00	ok	3,00	ok
299248	3,00	l'architecture respecte la donnée, évitez d'inclure des .h dans d'autres.h (seulement dans les .cc)	3,00	encapsulation respectée et interfaces externalisées, bien joué!
299274	2,00	[A3] tools.h includes constantes.h making it dependant of variables that are specific to this pproject -> tools should be independant	3,00	Excellent encapsulation.
299290	3,00	ok	3,00	Ok
299391	3,00	OK	3,00	Good class encapsulation. Warning thugh, you have a constructor defined in noeud.h (Quartier) which should be defined in noeud.cc, even if it is 2 lines, we won't deduct points this time :)
299450	3,00	Votre architecture respecte la donnée.	3,00	Bonne encapsulation. Attention néanmoins au getter qui retourne un pointeur sur un noeud dans votre classe ville. Il pourra être suceptible de la briser. Ok car vous ne le modifiez pas mais cela pourrait vous pénalisez si vous ne le rendez pas constant.
299579	3,00	ok	3,00	ok

299585	1,00	[A1] projet.cc include error.h, constante.h, tools.h, noeud.h et il y a deux fois l'include de ville.h. [A3] tools.h include constante.h, ce module doit etre totalement independant.	3,00	OK.
299871	2,00	Bonne hierchachie des modules mise à part : [A1]projet.cc:29 : l'instance de ville doit être declarée dans une fonction du module ville, dans votre projet, elle est déclarée dans la fonction main. Définissez une fonction du module ville qui se charge d'instancier la ville et ensuite appeller les méthodes de la classe.	3,00	Bravo très bonne encapsulation et bonne utilisation de l'héritage
299885	3,00	en incluant noeud.h dans ville.h puis ville.h dans projet.cc c'est comme si noeud.h était inclus dans projet.cc, pas de pénalité car vous ne l'utilisez pas	3,00	encapsulation respectée et interfaces externalisées, bien joué!
299971	2,00	[A1] projet ne doit pas include constantes.h , essayez de ne pas inclure des .h dans d'autres .h ni d'utiliser using namespace std (dans les .h)	3,00	encapsulation respectée et interfaces externalisées, bien joué!
300036	3,00	ok	3,00	ok
300038	2,00	Bon respect de l'architecture en général par contre : [A3]tools.cc:17,40 : Le module tools fait appel aux fonctions d'erreurs reliées au projet. Ce module ne doit pas avoir de notion de noeud ou d'erreurs de superposition de noeud et devrait pouvoir être réutilisé pour d'autres applications indépendantes.	2,00	La classe de noeuds est définie correctement [C1]noeud.cc:138,139 : Warning : Il est préférable de définir une classe pour respresenter la ville avec un attribut contenant tous les noeuds au lieu de declarer des variables statiques. [C2]noeud.h:42,52,61: Les constructeurs sur plusieurs lignes doivent être définis dans noeud.cc.
300048	2,00	[A1] projet.cc inclut error.h	0,00	[C2] les différents constructeurs de Noeud et celui de Ville ne tiennent pas sur la ligne de déclaration de la fonction
300086	2,00	[A1] main.cc calls error::success() ; argc, argv should be treated by main.cc, not by city	3,00	Ok

300089	0,00		0,00	
300148	2,00	[A1]projet.cc:21 : l'instance de ville doit être déclarée dans une fonction du module ville, dans votre projet, elle est déclarée dans la fonction main. Définissez une fonction du module ville qui se charge d'instancier la ville et ensuite appeler les méthodes de la classe. Hormis cela bonne architecture / bon module tools	3,00	Bonne encapsulation
300166	2,00	[A1] projet ne doit pas include constantes.h , essayez de ne pas inclure des .h dans d'autres .h ni d'utiliser using namespace std (dans les .h)	3,00	encapsulation respectée et interfaces externalisées, bien joué!
300247	3,00	OK	0,00	[C2] ville.h: class Ville interface is not externalized/ tools.h: class Segment and class Cercle interface not externalized/ noeud.h: class Noeud interface not externalized.
300259	3,00	Bravo pour la bonne organization des modules et respect de l'architecture	3,00	Bonne Encapsulation. Bonne séparation du module ville avec les fonctions dans l'espace nom non-nommé.
300299	3,00	Ok	3,00	Ok
300370	3,00	OK	3,00	OK
300371	3,00	OK	3,00	OK
300454	3,00	OK	3,00	OK

300780	3,00	Bonne hiérarchie, bravo !	3,00	La classe Ville est vide, donc inutile. Sinon la classe Noeud est correct. Peut-être clarifier la différence entre le namespace noeud et la classe Noeud.
300840	3,00	[A1] projet.cc includes noeud.h and error.h	3,00	Ok
300891	3,00	Vous respectez parfaitement l'architecture demandée !	3,00	la classe Noeud est public (la classe est définie dans noeud.h) tandis que la classe Ville est privée (dans ville.cc). C'est un choix possible. Vos classes sont sinon correctes !
300899	2,00	[A1] projet.cc includes nodes.h & tools.h	3,00	ok
300914	2,00	[A1] Class projet.cc includes tools.h and noeud.h, but should only include ville.h as access to the Ville module. Otherwise architecture is correct.	3,00	Good class encapsulation.
300987	2,00	[A1] projet.cc includes error.h et constante.h	0,00	[C1] 14 projet.cc: declaration of an instance of class Ville in the main, it breaks the encapsulation/ 11,12 ville.h; 19 noeud.h: attributes in public. [C2] noeud.h: class Noeud interface is not externalized.
301070	3,00	ok	3,00	ok

301271	3,00	OK	1,00	[C2] 31-38 Noeud.h, 23-30 Ville.h : for both the constructor definition is not externalized and doesn't hold on one line.
301295	2,00	[A1]projet.cc includes noeud.h	3,00	warning: the functions of ville.cc that are not declared in ville.h should be declared in the unnamed namespace of ville.cc
301319	3,00	ok	3,00	ok
301366	3,00	ok	3,00	ok

301407	3,00	noeud.h est inclu dans projet.cc et ne le devrait pas. Mais vous n'êtes pas pénalisé car vous ne l'utilisez pas. Utilisez peut-être des namespace pour que l'on puisse identifier les orgines des fonctions (par exemple: lecture et decodage_ligne)	3,00	Vous n'avez pas de classe ville, mais la classe Noeud est bien construite !
301438	3,00	Good architecture.	3,00	Very good encapsulation, everything that should be constant is.
301446	3,00	noeud.h est inclu dans projet.cc et ne le devrait pas. Mais vous n'êtes pas pénalisé car vous ne l'utilisez pas. Utilisez peut-être des namespace pour que l'on puisse identifier les orgines des fonctions (par exemple: lecture et decodage_ligne)	3,00	Vous n'avez pas de classe ville, mais la classe Noeud est bien construite !
301450	2,00	[A1] projet includes error.h ; le module tools ne doit pas include constantes.h mais pas de pénalité car non utilisé	1,00	[C1] noeud.cc L20,21 il est strictement interdit d'avoir des attributs publics, à corriger
301482	2,00	[A1]projet.cc includes noeud.h	3,00	warning: the functions of ville.cc that are not declared in ville.h should be declared in the unnamed namespace of ville.cc
301496	3,00	ok	3,00	encapsulation respectée et interfaces externalisées, bien joué!

301607	3,00	Ok	2,00	[C1] ville.h, noeud.h : vous commettez une erreur importante : vous définissez vos classes dans les .cc. Par conséquent vos classes ne montrent aucune interface à l'extérieur et sont inutilisables, ce qui vous oblige à des artifices étranges. Dans noeud.h vous définissez les classes Link et Noeud, et vous les re-définissez dans le .cc, ce qui vous a forcé à mettre les définitions de Noeud.h dans un namespace noeud pour que le compilateur accepte de compiler. Et ce qui vous force, dans le .cc, à utiliser les méthodes de Noeud en faisant noeud::Noeud::methode, alors que Noeud::methode aurait suffi. Warning : vous définissez des fonctions init en dehors de toute classe au lieu d'utiliser les constructeurs de ces classes, faits pour cela. Warning enfin : vous ne définissez aucun structure Cercle, Point ou Ligne dans votre tools, ce qui risque de vous compliquer sévèrement la tâche pour les prochains rendus.
301653	1,00	[A2] Class ville is declared in main in order to call the lecture function. This is a violation of architecture and you should instead have a function in the ville module that does this for you. [A1] : projet.cc includes tools.h and constantes.h	3,00	OK.
301707	3,00	Warning : Vous déléguez la lecture des données brutes du fichier jusque dans le module Noeud. Or seul le module ville doit lire ces données. A corriger pour le prochain rendu	3,00	Ok
301720	2,00	[A1] projet.cc includes noeud.h	3,00	Ok
301765	3,00	Bonne architecture mais ajoutez les define de gardes dans le fichier ville.h	3,00	La classe Noeud qui est définie dans le .h n'est jamais utilisée hors du module noeud. Est-ce voulu ? (Pas obligé de changer car ce n'est pas faux) Rien à redire sinon.
301786	3,00	Warning : Vous déléguez la lecture des données brutes du fichier jusque dans le module Noeud. Or seul le module ville doit lire ces données. A corriger pour le prochain rendu	3,00	Ok

301816	3,00	OK	2,00	la classe lien hérite d'une struc donc les attributs hérités sont publics, ce qui est interdit pour une classe selon nos conventions
301834	1,00	[A1] projet.cc includes everything, but should only include access to Ville module -> ville.h [A3] tools module is not independant of projet since it includes constantes.h, which are specific to this project.	2,00	[C1] Ville.cc contains global variables which violates encapsulation. Care because your class Noeud has a getter that returns the actual list of links (not a constant reference) which is the same as setting this to public which is not legal in your guidelines.
301837	2,00	[A1] projet.cc includes noeud.h	3,00	Ok
301850	1,00	[A1] projet.cc includes everything, but should only include access to Ville module -> ville.h [A3] tools module is not independant of projet since it includes constantes.h, which are specific to this project.	2,00	[C1] Ville.cc contains global variables which violates encapsulation. Care because your class Noeud has a getter that returns the actual list of links (not a constant reference) which is the same as setting this to public which is not legal in your guidelines.
301863	2,00	[A2] : La lecture doit se produire dans le module ville. Votre module ville ne sert actuellement à rien, puisque tout se passe dans Noeud. La ville ne peut même pas avoir accès à ses propres quartiers sans passer par le module Noeud	2,00	[C2] : Les constructeurs doivent être implémentés dans le .cc, sauf s'ils tiennent sur une ligne. Ce n'est pas le cas de celui de Noeud (noeud.h: 32) (même si vous vous êtes débrouillés pour que tout soit sur une ligne, ça devrait être sur plusieurs) Warning : la fonction lecture de Ville est inutilement déclarée static. Ou plutôt, la fonction file_to_read qui ne sert qu'à créer une instance de ville pour appeler lecture est inutile puisque vous aviez déclaré lecture statique et pouviez, depuis le main, écrire Ville::lecture(), et tout allait bien
301873	3,00	ok	3,00	ok
301900	3,00	Good architecture.	3,00	Very good encapsulation, everything that should be constant is.
301927	3,00	Architecture très bien respectée!	3,00	Interfaces claires et propres avec des fonctions bien externalisées.

301940	2,00	[A3] tools.h includes constantes.h making it dependant of variables that are specific to this pproject -> tools should be independant	3,00	Excellent encapsulation.
301950	0,00		0,00	
301964	3,00	ok	3,00	ok
302023	3,00	Architecture respectée, bravo !	3,00	L'implémentation et l'utilisation de classes dans votre projet est correcte mais les lignes avec le nom de la classe sont inutiles dans votre cas (noeud.h:7-8, par exemple). Vous pourriez aussi mettre la fonction lecture de ville dans la classe Ville.
302038	2,00	[A3] variable dist_2quartiers est déclarée dans tools.cc ligne 13. Très bien, l'organization du projet suit clairement les directives et definit une semantique logique par rapport à l'application. Cependant meme si le module tools.cc n'utilise pas les modules au dessus une variable dist_2quartier y est définie, etant donné que le module tools ne devrait avoir aucune notion de noeud/quartier il serait préférable de renommer cette variable à `distance` par exemple.	3,00	Bonne encapsulation bravo ! Warning : Un petit détail: le vecteur de noeud "graphe" défini dans l'espace non-nommé est utilisé pour superviser les noeuds ajouté précédemment. Il serait préférable de définir une classe representant la ville qui contient une liste de noeud privée en tant qu'attribut. (cela simplifiera surement les prochaines taches du projet)

302153	3,00	OK	3,00	OK
302297	3,00	ok	3,00	ok
302394	3,00	ok	3,00	ok
302397	3,00	Warning: tools.cc 20 le module tools ne doit pas connaître le concept de noeud. De ce fait, ce mot ne devrait pas se trouver dans le nom de vos fonctions. Cela pourrait vous pénaliser pour les prochains rendus.	2,00	[C2] noeud.h 29,32 les définitions des getters et de la liste d'initialisation du constructeur sont uniquement acceptées si elles tiennent sur une seule ligne. Warning: [C1] noeud.h 28 Vous donnez directement un accès à un attribut privé en retournant un pointeur. Ok car vous n'utilisez pas la fonction pour le modifier mais veuillez à corriger ça.
302457	3,00	ok	3,00	perfect

302508	1,00	<p>[A1] projet.cc contains an instance of Town, it should only call a function from the Town Module.</p> <p>[A3] tools.h is not completely independant of the project since it contains constantes.h, file specific to this project.</p>	3,00	<p>Care since your getters return the actual lists and not constant references which is the same as setting your list as public (illegal in our guidelines). You do the same for returning structures which contain public variables.</p> <p>Otherwise good job!</p>
302576	3,00	<p>en incluant noeud.h dans ville.h puis ville.h dans projet.cc c'est comme si noeud.h était inclus dans projet.cc, pas de pénalité car vous ne l'utilisez pas</p>	3,00	<p>encapsulation respectée et interfaces externalisées, bien joué!</p>
302680	3,00	<p>l'architecture respecte la donnée</p>	3,00	<p>encapsulation respectée et interfaces externalisées, bien joué!</p>
302699	2,00	<p>[A1] l'instance de ville ne doit pas être déclarée dans projet.cc mais dans le module ville; évitez d'inclure des .h dans d'autres.h (seulement dans les .cc). le module tools est indépendant et ne doit pas inclure error et constantes, pas de pénalité car non utilisés</p>	3,00	<p>encapsulation respectée et interfaces externalisées, bien joué!</p>

302808	1,00	[A1] projet.cc include error.h, constante.h, tools.h, noeud.h et il y a deux fois l'include de ville.h. [A3] tools.h include constante.h, ce module doit etre totalement indépendant.	3,00	OK.
302875	2,00	[A1] projet.cc inclut error.h	0,00	[C2] les différents constructeurs de Noeud et celui de Ville ne tiennent pas sur la ligne de déclaration de la fonction
302888	3,00	Ok	2,00	[C1] ville.h, noeud.h : vous commettez une erreur importante : vous définissez vos classes dans les .cc. Par conséquent vos classes ne montrent aucune interface à l'extérieur et sont inutilisables, ce qui vous oblige à des artifices étranges. Dans noeud.h vous définissez les classes Link et Noeud, et vous les re-définissez dans le .cc, ce qui vous a forcé à mettre les définitions de Noeud.h dans un namespace noeud pour que le compilateur accepte de compiler. Et ce qui vous force, dans le .cc, à utiliser les méthodes de Noeud en faisant noeud::Noeud::methode, alors que Noeud::methode aurait suffi. Warning : vous définissez des fonctions init en dehors de toute classe au lieu d'utiliser les constructeurs de ces classes, faits pour cela. Warning enfin : vous ne définissez aucun structure Cercle, Point ou Ligne dans votre tools, ce qui risque de vous compliquer sévèrement la tâche pour les prochains rendus.
302910	2,00	[A1] tu fait plus que necessaire dans le main , pas besoin d afficher le contenu de la ville	3,00	ok
302939	3,00	ok	3,00	ok
302953	2,00	[A3] le module tool est general , il ne doit pas utiliser constante.h qui est specifique au projet	3,00	ok

303000	3,00	évitez d'inclure des .h dans d'autres.h (seulement dans les .cc) sinon le module noeud par exemple est indirectement inclus dans projet	3,00	Ok
303084	3,00	OK	3,00	Excellent class encapsulation. Warning though, you have a constructor defined in noeud.h (Quartier) which should be defined in noeud.cc, even if it is 2 lines, we won't deduct points this time :)
303114	3,00	ok	3,00	perfect
303207	1,00	[A1] class projet.cc includes and uses variables from error.h, this violates architecture. Ville must have the responsibility of calling error::success if it succeeds in creating an instance, not the higher module. [A3] tools module accesses constantes.h, these are related to this project and thus make so that tools module is not independant and reusable for other projects. This module should be entirely independent and not rely on project specific data. Watchout for includes that are not being used.	3,00	Excellent class encapsulation!
303227	1,00	[A1] class projet.cc includes error.h, this violates architecture. [A3] tools module accesses constantes.h, these are related to this project and thus make so that tools module is not independant and reusable for other projects. This module should be entirely independent and not rely on project specific data. Watchout for includes that are not being used.	3,00	Good class encapsulation!
303778	3,00	OK	0,00	[C2] ville.h: class Ville interface is not externalized/ tools.h: class Segment and class Cercle interface not externalized/ noeud.h: class Noeud interface not externalized.
309996	3,00	ok	3,00	ok

310030	3,00	Warning why do you have lecture.cc & projet.cc with the same content ?	3,00	ok
310074	3,00	Ok	3,00	Ok
310075	3,00	Votre architecture suit parfaitement la donnée.	3,00	Bonne encapsulation
310087	3,00	Ok	3,00	Ok
310104	3,00	ok	3,00	ok
310106	3,00	ok	3,00	ok
310108	0,00		0,00	
310128	3,00	Vous incluez noeud dans dans project.cc, ce qu'il ne faut pas faire. Vous ne l'utilisez pas donc pas de pénalité. En outre, vous utilisez des fonctions comme read_file pour faire l'intermédiaire avec votre objet Ville, pourquoi ne pas utilisez la lecture de fichier dans le constructeur de ville ? (vous n'êtes pas obligé de changer: ce que vous faites est juste). Sinon, vous respectez la hiérarchie demandée !	3,00	Parfait. Vous utilisez même des const pour vous contraindre d'avantage dans la suite du projet. Excellente habitude.

310261	3,00	Ok	1,00	[C1] ville.h: 14, 15, noeud.h: 33 : Attribut public pour noeud, et vos getters de ville retournent des références non-constantes utilisées par Noeud pour remplir les vecteurs de Noeud de ville. Cela revient à déclarer ces vecteurs publics, car une fois la référence obtenue par une classe externe, plus rien ne peut l'empêcher de modifier entièrement le vecteur de Noeud, sans aucun contrôle de la part de Ville ! Warning : les méthodes que vous déclarez dans les .cc afin qu'elles ne soient pas accessibles à d'autres devraient être des méthodes private !
310271	3,00	Ok	3,00	Warning : vous envoyez du lourd sur tout et votre respect de l'orienté-objet et de l'encapsulation est au top, donc je me permets de couper des cheveux en 4 : votre fonction removeNode en dehors de toute classe dans le module Noeud devrait être une fonction de Ville. A corriger !
310276	3,00	Warning: tools.h 17 le module tools n'a pas connaissance du concept de lien. De ce fait ce mot ne devrait pas se trouver dans le nom de ses méthodes.	3,00	Vos interfaces sont claires et bien externalisées et c'est une bonne pratique de rendre les getters constants. Excellent!
310279	2,00	[A3] dist_min est une constante spécifique au projet , elle ne devrait pas se trouver dans le module tool qui est general	2,00	Warning : supprimez les prototypes que vous n'utilisez pas. [C2] dans le .h tu ne doit pas mettre de définition de fonction , juste les prototypes . ex : Noeud::ajouter_lien
310295	3,00	Bon respect de l'architecture et donc bonne hiérarchie des modules Bravo !	3,00	Très Bonne encapsulation et respect des directives pour les classes Bravo
310349	2,00	[A2] : La lecture doit se produire dans le module ville. Votre module ville ne sert actuellement à rien, puisque tout se passe dans Noeud. Warning : vous n'avez pas nettoyé vos includes, vous incluez iostream dans tools (en plus sans chevrons, ce qui est inutile. Pensez à n'inclure que les fichiers nécessaires dans chaque module	0,00	[C1] : la classe Noeud dispose d'un getter qui renvoie une référence sur l'entièreté du vecteur voisin. Cela revient à déclarer ce vecteur comme attribut public. [C2] : noeud.h, ville.h : les constructeurs doivent être définis dans le .cc, un .h ne contient que des prototypes ou très éventuellement des fonctions qui tiennent sur une ligne (genre les constructeurs des sous-classes de Node que vous avez très bien définis) ! Warning : le vecteur qui contient l'entièreté des noeuds (tab_n) doit être défini dans ville !
310409	3,00	Ok	3,00	Ok
310410	3,00	ok	3,00	ok
310447	3,00	La hiérarchie est parfaitement respectée. Au niveau de la fonctionnalité de main.cc, la ligne if(argc != 2); ne fait rien => à corriger. Vous pourriez aussi mettre les fonctions de tools dans un namespace tools ou un préfixe "à la C": par exemple, d_centres (dont on ignore au nom l'origine) deviendrait tools::d_centres (avec le namespace) ou tools_d_centres ("à la C").	3,00	Vos classes sont bien faites (notamment, on comprend directement comment utiliser les classes avec uniquement le .h) ! Bravo !

310572	2,00	[A2] : La lecture doit se produire dans le module ville. Votre module ville ne sert actuellement à rien, puisque tout se passe dans Noeud. Warning : vous n'avez pas nettoyé vos includes, vous incluez iostream dans tools (en plus sans chevrons, ce qui est inutile. Pensez à n'inclure que les fichiers nécessaires dans chaque module	0,00	[C1] : la classe Noeud dispose d'un getter qui renvoie une référence sur l'entièreté du vecteur voisin. Cela revient à déclarer ce vecteur comme attribut public. [C2] : noeud.h, ville.h : les constructeurs doivent être définis dans le .cc, un .h ne contient que des prototypes ou très éventuellement des fonctions qui tiennent sur une ligne (genre les constructeurs des sous-classes de Node que vous avez très bien définis) ! Warning : le vecteur qui contient l'entièreté des noeuds (tab_n) doit être défini dans ville !
310573	3,00	Architecture respectée, Evitez les includes inutiles dans des .h (par exemple dans votre ville.h)	1,00	[C1] noeud.cc 11-12 vous déclarez deux variables globales! Elles ne sont acceptées dans cette portée que si elles sont statiques ou dans l'espace de nom non nommé
310576	3,00	ok	3,00	ok
310608	3,00	Utilisez peut-être des namespace pour que l'on puisse identifier les origines des fonctions (par exemple readfile du module ville). Sinon l'architecture est correcte. C'est bien !	3,00	Les classes sont utilisées de manière correctes, bravo !
310616	3,00	Ok	3,00	Ok
310635	3,00	ok	3,00	Ok
310672	3,00	OK	3,00	OK
310674	3,00	Warning: tools.h 17 le module tools n'a pas connaissance du concept de lien. De ce fait ce mot ne devrait pas se trouver dans le nom de ses méthodes.	3,00	Vos interfaces sont claires et bien externalisées et c'est une bonne pratique de rendre les getters constants. Excellent!

310741	3,00	Ok	3,00	Ok
310786	3,00	La hiérarchie est parfaitement respectée. Au niveau de la fonctionnalité de main.cc, la ligne if(argc != 2); ne fait rien => à corriger. Vous pourriez aussi mettre les fonctions de tools dans un namespace tools ou un préfixe "à la C": par exemple, d_centres (dont on ignore au nom l'origine) deviendrait tools::d_centres (avec le namespace) ou tools_d_centres ("à la C").	3,00	Vos classes sont bien faites (notamment, on comprend directement comment utiliser les classes avec uniquement le .h) ! Bravo !
310832	3,00	Warning : vous n'avez pas nettoyé vos includes, vous utilisez ifstream dans projet, ce qui est inutile, et ce n'est pas le seul. Pensez à n'inclure que les fichiers nécessaires dans chaque module	3,00	Warning : vous ne devez pas dans votre classe Ville avoir des getters qui permettent de récupérer l'entièreté du vecteur de Noeud. Heureusement, vos getters n'en passent qu'une copie et non une référence (ce qui vous sauve le point), mais c'est tout de même à éviter
310847	3,00	Ok	3,00	Ok
310863	2,00	projet.cc 17,18,20,21 Projet a pour seule mission de traiter les argc/argv et de se connecter au modèle via une seule fonction de ville. Vous déclarez ici des objets pour appeler des fonctions.	3,00	Warning ville.cc 95 retourner un pointeur rend les attributs de votre tableau privé publics. Ok car vous ne les modifiez pas mais veillez à les rendre constants afin de ne pas briser l'encapsulation.
310978	2,00	[A2] noeud.cc inclus dans ville.h	3,00	ok
310986	3,00	Warning why do you have lecture.cc & projet.cc with the same content ?	3,00	ok
311013	3,00	Warning : remove useless includes , tools should not include constants , as constants are specifics to the project and tool is general	3,00	ok
311043	3,00	OK	2,00	[C1] 15 projet.cc: declaration of an instance of class Ville in the main, it breaks the encapsulation.

311105	3,00	Ok	3,00	Ok
311112	3,00	ok	3,00	ok
311155	2,00	[A3] le module tools est indépendant et ne doit pas faire intervenir les constantes du projet	3,00	encapsulation respectée et interfaces externalisées, bien joué!
311236	2,00	[A2] : La lecture doit se produire dans le module ville. Votre module ville ne sert actuellement à rien, puisque tout se passe dans Noeud. La ville ne peut même pas avoir accès à ses propres quartiers sans passer par le module Noeud	2,00	[C2] : Les constructeurs doivent être implémentés dans le .cc, sauf s'ils tiennent sur une ligne. Ce n'est pas le cas de celui de Noeud (noeud.h: 32) (même si vous vous êtes débrouillés pour que tout soit sur une ligne, ça devrait être sur plusieurs) Warning : la fonction lecture de Ville est inutilement déclarée static. Ou plutôt, la fonction file_to_read qui ne sert qu'à créer une instance de ville pour appeler lecture est inutile puisque vous aviez déclaré lecture statique et pouviez, depuis le main, écrire Ville::lecture(), et tout allait bien
311252	3,00	ok	3,00	Warning Point struct in tool.h , wierd double declaration and why another typedef w/ same name
311385	2,00	[A1] mauvaise encapsulation , ta ville devais etre dans son module , pas dans le main , et le main devrais utiliser seulemet 1 methode du module ville.	3,00	ok
311390	3,00	ok	3,00	ok
311574	2,00	[A1] main.cc n'inclut pas city mais test	3,00	ok
311644	2,00	[A3] tools.cc 48 l'utilisation d'une constante telle que "dist_min" (constante spécifique à ce projet) ne devrait pas se trouver dans un module général tel que tools	3,00	Bonne encapsulation
311730	3,00	Ok	3,00	Ok

311913	2,00	[A1] projet.cc inclut error.h et noeud.h	2,00	[C2] le constructeur de Noeud ne tient pas sur une seule ligne
311916	2,00	[A1] projet doit seulement inclure ville et pas error ni node	3,00	encapsulation respectée et interfaces externalisées, bien joué!
311970	3,00	Warning : vous vous en êtes bien rendus compte, mais en effet votre classe Ville actuellement ne sert à rien. Elle sera nécessaire dans le futur, le vecteur de Noeud devra en être un attribut, et certaine fonction de détection d'erreur devront donc nécessairement être évaluées dans Ville	3,00	Ok
311980	2,00	[A3] tools.cc 48 l'utilisation d'une constante telle que "dist_min" (constante spécifique à ce projet) ne devrait pas se trouver dans un module général tel que tools	3,00	Bonne encapsulation
312066	3,00	OK	2,00	[C1] 15 projet.cc: declaration of an instance of class Ville in the main, it breaks the encapsulation.
312078	2,00	Bonne architecture en général cependant: [A1] projet.cc:9 : l'instance de ville doit être déclarée dans une fonction du module ville, dans votre projet, elle est déclarée dans la fonction main. Définissez une fonction du module ville qui se charge d'instancier la ville et ensuite appeller les méthodes de la classe.	2,00	[C2]noeud.h:16-25. Le constructeur de noeud est défini dans l'interface sur plusieurs lignes, il faut le définir dans le fichier source noeud.cc. Hormis cela bonne encapsulation bravo
312106	3,00	ok	3,00	ok
312120	3,00	OK	3,00	OK
312121	2,00	[A1] projet.cc includes noeud.h	3,00	Ok
312280	2,00	[A1] projet.cc inclut error.h et noeud.h	2,00	[C2] le constructeur de Noeud ne tient pas sur une seule ligne

312282	3,00	Warning : vous n'avez pas nettoyé vos includes, vous utilisez ifstream dans projet, ce qui est inutile, et ce n'est pas le seul. Pensez à n'inclure que les fichiers nécessaires dans chaque module	3,00	Warning : vous ne devez pas dans votre classe Ville avoir des getters qui permettent de récupérer l'entièreté du vecteur de Noeud. Heureusement, vos getters n'en passent qu'une copie et non une référence (ce qui vous sauve le point), mais c'est tout de même à éviter
312296	3,00	ok	3,00	ok
312367	2,00	[A1] projet doit seulement inclure ville et pas error ni node	3,00	encapsulation respectée et interfaces externalisées, bien joué!
312453	3,00	Votre architecture respecte la donnée	3,00	Très bonne encapsulation et bon usage des namespaces!
312460	2,00	[A2] noeud.cc inclus ville.h	3,00	ok
312577	3,00	Ok	1,00	[C1] ville.h: 14, 15, noeud.h: 33 : Attribut public pour noeud, et vos getters de ville retournent des références non-constantes utilisées par Noeud pour remplir les vecteurs de Noeud de ville. Cela revient à déclarer ces vecteurs publics, car une fois la référence obtenue par une classe externe, plus rien ne peut l'empêcher de modifier entièrement le vecteur de Noeud, sans aucun contrôle de la part de Ville ! Warning : les méthodes que vous déclarez dans les .cc afin qu'elles ne soient pas accessibles à d'autres devraient être des méthodes private !
312620	3,00	Warning : elimine tes includes inutile , tools ne doit pas include constates.h car c est un module generale et constantes.h est specifique au projet	2,00	[C2] pas de definition dans les .h , juste des prototypes. La seule exception sont les getteurs si ils tiennent sur une ligne
312648	2,00	[A1] Class projet.cc includes tools.h and noeud.h, but should only include ville.h as access to the Ville module. Otherwise architecture is correct.	3,00	You have an empty class Ville, if it remains like this in the following parts of the project you can just remove this class declaration. Otherwise good job!
312656	3,00	ok	3,00	ok

312737	3,00	Bonne architecture mais ajoutez les define de gardes dans le fichier ville.h	3,00	La classe Noeud qui est définie dans le .h n'est jamais utilisée hors du module noeud. Est-ce voulu ? (Pas obligé de changer car ce n'est pas faux) Rien à redire sinon.
312757	3,00	Bon respect de l'architecture et donc bonne hiérarchie des modules Bravo ! conseil pour tools.cc: pensez à ajouter des structs pour représenter les points et vecteurs, ça évite les fonctions à 4 paramètres et rend le code plus lisible.	2,00	[C1]ville.h:12,13 : les getters getIles() et getLiens() retournent des références non constantes aux vecteurs îles et liens, ce qui revient au même qu'avoir ces attributs public.
312854	3,00	Ok	3,00	Ok
312920	2,00	[A1] mauvaise encapsulation, ta ville devait être dans son module, pas dans le main, et le main devrait utiliser seulement 1 méthode du module ville.	3,00	ok
312936	3,00	Bonne architecture bravo, vos modules utilisent le strict nécessaire	3,00	- Bonne encapsulation bravo
312983	3,00	ok	3,00	ok
313001	3,00	Ok	3,00	ok
313218	3,00	Warning : la classe Noeud ne doit pas avoir à parser des données brutes lues depuis le fichier. C'est à Ville de le faire, puis d'appeler un constructeur de Noeud avec les bonnes valeurs. A corriger	3,00	Warning : passer à chaque noeud un pointeur sur l'ensemble des noeuds de la ville est une rupture directe de l'encapsulation, puisque vous donnez accès à une classe autre que ville à un attribut private. Cela doit être corrigé
313312	3,00	OK	2,00	[C2] 50-52 nodes.h: the constructor definition is not externalized and doesn't hold on one line.

313318	3,00	ok	3,00	ok
313415	3,00	Votre architecture suit parfaitement la donnée.	3,00	Bonne encapsulation
313705	3,00	Bonne hiérarchie, bravo !	3,00	La classe Ville est vide, donc inutile. Sinon la classe Noeud est correct. Peut-être clarifier la différence entre le namespace noeud et la classe Noeud.
314353	2,00	[A1] Class projet.cc includes tools.h and noeud.h, but should only include ville.h as access to the Ville module. Otherwise architecture is correct.	3,00	You have an empty class Ville, if it remains like this in the following parts of the project you can just remove this class declaration. Otherwise good job!
314377	3,00	Votre architecture respecte la donnée	3,00	Très bonne encapsulation et bon usage des namespaces!
314443	2,00	[A3] variable dist_2quartiers est déclarée dans tools.cc ligne 13. Très bien, l'organization du projet suit clairement les directives et definit une semantique logique par rapport à l'application. Cependant meme si le module tools.cc n'utilise pas les modules au dessus une variable dist_2quartier y est définie, etant donné que le module tools ne devrait avoir aucune notion de noeud/quartier il serait préférable de renommer cette variable à `distance` par exemple.	3,00	Bonne encapsulation bravo ! Warning : Un petit détail: le vecteur de noeud "graphe" défini dans l'espace non-nommé est utilisé pour superviser les noeuds ajouté précédemment. Il serait préférable de définir une classe representant la ville qui contient une liste de noeud privée en tant qu'attribut. (cela simplifiera surement les prochaines taches du projet)

314456	3,00	OK	2,00	[C2] 50-52 nodes.h: the constructor definition is not externalized and doesn't hold on one line.
314470	2,00	[A1] projet.cc includes error.h	0,00	[C1] noeud.cc 9,10 [C2] all methods of classes are defined in the .h
314472	2,00	[A3] tool doit etre independand de ville/noeud , c est un module generale	2,00	[c1] il est interdit d utiliser des variables globales . Warning les definitions des getteur doivent tenir sur une ligne si vous les laissez dans le .h .
314477	2,00	projet.cc 17,18,20,21 Projet a pour seule mission de traîter les argc/argv et de se connecter au modèle via une seule fonction de ville. Vous déclarez ici des objets pour appeler des fonctions.	3,00	Warning ville.cc 95 retourner un pointeur rend les attributs de votre tableau privé publics. Ok car vous ne les modifiez pas mais veillez à les rendre constants afin de ne pas briser l'encapsulation.

314483	2,00	[A3] tools.h includes constantes.h making it dependant of variables that are specific to this pproject -> tools should be independant	2,00	[C2] Noeud.h contains a constructor that is not externalized and does not fit in one line. Warning, your getters return the actual vectors instead of constant references which is the same as setting these to Public, illegal in the guidelines. You also define an empty class Ville, is this necessary?
314495	3,00	ok	3,00	ok
314504	3,00	Architecture respectée, Evitez les includes inutiles dans des .h (par exemple dans votre ville.h)	1,00	[C1] noeud.cc 11-12 vous déclarez deux variables globales! Elles ne sont acceptées dans cette portée que si elles sont statiques ou dans l'espace de nom non nommé
314754	2,00	Bonne architecture en général cependant: [A1] projet.cc:9 : l'instance de ville doit être déclarée dans une fonction du module ville, dans votre projet, elle est déclarée dans la fonction main. Définissez une fonction du module ville qui se charge d'instancier la ville et ensuite appeller les méthodes de la classe.	2,00	[C2]noeud.h:16-25. Le constructeur de noeud est défini dans l'interface sur plusieurs lignes, il faut le définir dans le fichier source noeud.cc. Hormis cela bonne encapsulation bravo

314886	3,00	ok	3,00	Ok
314939	3,00	Bon respect de l'architecture et donc bonne hierarchie des modules Bravo !	3,00	Très Bonne encapsulation et respect des directives pour les classes Bravo
314962	3,00	Vous respectez parfaitement l'architecture demandée !	3,00	la classe Noeud est public (la classe est définie dans noeud.h) tandis que la classe Ville est privée (dans ville.cc). C'est un choix possible. Vos classes sont sinon correctes !

315014	3,00	OK	3,00	OK
315019	2,00	[A3] tools.h includes constantes.h making it dependant of variables that are specific to this pproject -> tools should be independant	2,00	[C2] Noeud.h contains a constructor that is not externalized and does not fit in one line. Warning, your getters return the actual vectors instead of constant references which is the same as setting these to Public, illegal in the guidelines. You also define an empty class Ville, is this necessary?
315065	3,00	Ok	3,00	Ok
315120	3,00	OK	3,00	OK

315192	2,00	[A3] dist_min est une constante specifique au projet , elle ne devrai pas se trouver dans le module tool qui est general	2,00	Warning : supprimez les prototypes que vous n utilisez pas. [C2] dans le .h tu ne doit pas mettre de definition de fonction , juste les prototypes . ex : Noeud::ajouter_lien
315273	3,00	ok	3,00	ok
315334	2,00	[A1] projet.cc includes error.h	0,00	[C1] noeud.cc 9,10 [C2] all methods of classes are defined in the .h

315408	3,00	Warning : elimine tes includes inutile , tools ne doit pas include constates.h car c est un module generale et constantes.h est specifique au projet	2,00	[C2] pas de definition dans les .h , juste des prototypes. La seule exception sont les getteurs si ils tiennent sur une ligne
315430	3,00	Vous incluez noeud dans dans project.cc, ce qu'il ne faut pas faire. Vous ne l'utilisez pas donc pas de pénalité. En outre, vous utilisez des fonctions comme read_file pour faire l'intermédiaire avec votre objet Ville, pourquoi ne pas utilisez la lecture de fichier dans le constructeur de ville ? (vous n'êtes pas obligé de changer: ce que vous faites est juste). Sinon, vous respectez la hiérarchie demandée !	3,00	Parfait. Vous utilisez même des const pour vous contraindre d'avantage dans la suite du projet. Excellente habitude.

315536	3,00	Ok	3,00	Ok
315548	1,00	[A2] Class ville is declared in main in order to call the lecture function. This is a violation of architecture and you should instead have a function in the ville module that does this for you. [A1] : projet.cc includes tools.h and constantes.h	3,00	OK.
315567	3,00	ok	3,00	ok
315587	2,00	[A3] tool doit etre independand de ville/noeud , c est un module generale	2,00	[c1] il est interdit d utiliser des variables globales . Warning les definitions des getteur doivent tenir sur une ligne si vous les laissez dans le .h .

315651	3,00	Ok	3,00	Warning : vous envoyez du lourd sur tout et votre respect de l'orienté-objet et de l'encapsulation est au top, donc je me permets de couper des cheveux en 4 : votre fonction removeNode en dehors de toute classe dans le module Noeud devrait être une fonction de Ville. A corriger !
315788	3,00	OK	1,00	[C1] 14 projet.cc: declaration of an instance of class Ville in the main, it breaks the encapsulation. [C2] noeud.h: class Noeud interface is not externalized.
315824	3,00	ok	3,00	ok
315856	2,00	[A3] le module tools est indépendant et ne doit pas faire intervenir les constantes du projet	3,00	encapsulation respectée et interfaces externalisées, bien joué!
315930	3,00	[A1] projet.cc doit seulement contenir la fonction main(argc, argv[])	3,00	ok
315948	3,00	Bonne architecture bravo, vos modules utilisent le strict nécessaire	3,00	- Bonne encapsulation bravo
315974	3,00	ok	3,00	Warning Point struct in tool.h , wierd double declaration and why another typedef w/ same name
315983	3,00	OK	1,00	[C1] 14 projet.cc: declaration of an instance of class Ville in the main, it breaks the encapsulation. [C2] noeud.h: class Noeud interface is not externalized.
316049	3,00	Bon respect de l'architecture et donc bonne hierarchie des modules Bravo ! conseil pour tools.cc: pensez à ajouter des structs pour représenter les points et vecteurs, ça évite les fonctions à 4 parametres et rend le code plus lisible.	2,00	[C1]ville.h:12,13 : les getters getlles() et getLiens() retournent des références non constantes aux vecteurs iles et liens, ce qui revient au même qu'avoir ces attributs public.
316117	3,00	ok	3,00	ok
316132	2,00	[A1] main.cc n'inclut pas city mais test	3,00	ok

316258	3,00	Vous respectez la logique des architectures proposées mais vous rajoutez vos propres fichiers/modules. Evitez cela, surtout qu'ils ne sont pas nécessaire et leurs contenus pourraient être mis dans d'autres fichiers.	3,00	Très bonne classes, bien compréhensibles !
316306	3,00	ok	3,00	ok
316312	2,00	[A1]projet.cc:21 : l'instance de ville doit être déclarée dans une fonction du module ville, dans votre projet, elle est déclarée dans la fonction main. Définissez une fonction du module ville qui se charge d'instancier la ville et ensuite appeller les méthodes de la classe. Hormis cela bonne architecture / bon module tools	3,00	Bonne encapsulation
316333	3,00	Ok	3,00	ok
316368	2,00	[A3] le module tool est general , il ne doit pas utiliser constante.h qui est spécifique au projet	3,00	ok
316437	3,00	Ok	3,00	Ok
316518	3,00	Warning : remove useless includes , tools should not include constants , as constants are specifics to the project and tool is general	3,00	ok
316535	3,00	Warning : la classe Noeud ne doit pas avoir à parser des données brutes lues depuis le fichier. C'est à Ville de le faire, puis d'appeler un constructeur de Noeud avec les bonnes valeurs. A corriger	3,00	Warning : passer à chaque noeud un pointeur sur l'ensemble des noeuds de la ville est une rupture directe de l'encapsulation, puisque vous donnez accès à une classe autre que ville à un attribut private. Cela doit être corrigé
316536	3,00	Bravo pour la bonne organization des modules et respect de l'architecture	3,00	Bonne Encapsulation. Bonne séparation du module ville avec les fonctions dans l'espace nom non-nommé.
316830	3,00	Utilisez peut-être des namespace pour que l'on puisse identifier les orgines des fonctions (par exemple readfile du module ville). Sinon l'architecture est correcte. C'est bien !	3,00	Les classes sont utilisées de manière correctes, bravo !