

SCIPER	Code style	violation list	Comment on style (please note that penalty for indentation and line wrapping is at least in 2 places)
216913	2.00		L'indentation est impeccable, le code est agréable à lire et aéré, bravo ! N'hésitez pas à faire un peu plus de commentaire.
296190	2.00		Très bien dans l'ensemble. L'initialisation de variables dans une boucle n'est cependant pas une bonne pratique. Attention aux while(true), généralement on essaie des les éviter. Faites aussi attention aux if-else imbriqué, si il y a trop d'étage le code devient difficile à lire (fonction compare).
297806			
301678	1.50	[L11]83,243,428,433	Bon style, quelques petits problèmes d'indentation mais sinon très bien.
302064			
302461	1.50	[L14]147, 154, 376, ...	Code assez propre et bon style. Quelques problèmes au niveau de l'indentation de certains blocs de code. De plus le code n'est pas très consistant en terme d'espaces entre les lignes, ce qui est dommage et rend le code un peu moins lisible.
311144	2.00		Les conventions sont très bien respectées !
312271	1.50	[L22]239, 252, 256-257	Le code est globalement très bien indenté et le style est constant, malgré quelques erreurs d'alignement. Attention, la convention pour les noms des struct est qu'ils doivent commencer par une majuscule (E13).
314564	1.00	[L14]404 [L2]94, 269	Dans l'ensemble, conventions maîtrisées. Attention à l'indentation, une double indentation pour un bloc if ainsi qu'une indentation manquante (ligne 169). Attention également à la limite de 87 caractères qui a été dépassée 2 fois.
314664	1.50	[L14]11 ; ([L1]294)	Très beau style, dommage pour ces deux malheureuse indentations en trop, et la double indentation de votre structure. n'hésitez pas à séparer le typdef de la déclaration de votre structure pour faciliter la lecture.
315404	2.00		superbe
315678	2.00		Ajouter des espaces et des lignes vides rendrait le code plus facilement lisible, néanmoins le style est impeccable !
315801			
315905			
315933	1.00	[L1]38,63,95,...,300 ; [L2]93,228,261	Le code est globalement assez aéré et agréable à lire. Attention cependant à garder une indentation constante tout au long du programme. Les nom des variables et fonctions sont bien choisis et clairs, l'ajout de quelques commentaires donnerait encore un plus du côté de la clareté.
315980	2.00		Bon style , code bien lisible.
316105	2.00		Attention aux "constexpr" utilisés qui ne sont pas de bonnes pratiques, ainsi qu'à la limite de 87 caractère par ligne (dépassée à la ligne 162). Sinon, conventions de style maîtrisées.
316133			
316149	0.50	[L1]80,119,150 ; [L14]238 ; [L2] 36,178	Le style est dans l'ensemble très bien. Soyez cependant attentif au style de "{}" que vous utilisez, choisissez en un et gardez le pour tout le code, if, for, while, fonction, ... Soyez aussi attentif à respecter le nombre maximum de caractère par ligne, commentaires compris. N'hésitez pas pour ça à écrire vos instructions sur deux lignes si nécessaire. Qu'avez vous voulu faire à la ligne 238 ? Ne laissez pas d'instruction à moitié écrite, car ici elle ne pose pas de problème au compilateur car syntaxiquement juste, mais elle n'a probablement pas l'effet que vous vouliez.

316473			
316708	1.00	L[14]65, 66, 67, 68, 69 L[11]273	Les conventions de style sont globalement respectées, attention à la limite de 87 caractères qui a été dépassée à la ligne 103. Attention également à l'indentation.
324414	1.50	[L1]125,132,133,135	Le code est en général bien structuré. Attention à la cohérence dans le style d'accolades et d'indentation, et aux magic numbers (l.315,349,386). Le nom des variables doit être en minuscules (Succes, l.422)
324605	1.00	[L14]158:161 [L22]241,330	Quelques erreurs d'indentation mais dans l'ensemble le style est bon
324718	2.00		Le code est très bien structuré. L'utilisation du nom "mot" à la fois pour le nom d'une variable (de type "Mot") et d'un champ de cette même structure rend la lecture plus compliquée.
324806	2.00		Joli code, aéré (peut-être un peu trop des fois, une seule ligne vide suffit généralement) et agréable à lire. Attention, il y a une petite erreur d'indentation à la ligne 417. Le bloc de code avec les prototype mériterait peut-être d'être un peu espacé, en espaçant les fonctionnalités.
324815	1.50	[L22]145,148,204	Bon style de code, attention à l'alignement lorsqu'une ligne dépasse les 87 caractères de long. Aussi, l'utilisation des tabulations à la place des espaces rend l'indentation plus simple à gérer sur tout le code.
325002	1.00	[L1]101, 107, 166, 171 [R2]14	Une bonne indentation malgré un petit manque de cohérence rendant la lecture du code plus difficile. Déclarer AFFICHAGE en variable globale n'est pas une bonne pratique, ce genre de pratique aurait été adapté avec une classe possédant comme attribut le texte AFFICHAGE mais ça n'est pas dans le cadre de ce projet.
325420			
325828	1.00	[L2]32,99,151 ; [R2] errorStrings	Excellente indentation tout au long du programme, attention cependant à la dernière fonction qui est écrite différemment. Attention à ne pas mettre des exit(0) là où il n'y en a pas besoin car cela nuit à la lisibilité du code. Dommage de n'avoir pas géré les lignes trop longues, mis à part ça le code est compact et bien écrit. Une ligne vide de temps à autre ne serait pas la malvenue, les noms des fonctions sont très bien choisis. Attention également aux warnings mais on sent qu'avec un tout petit peu plus de travail ce code aurait pu être très bon
325856	1.00	[L1]297; [L14]176, 199; [L2]624; [L22]288, 294, 308, 332 ...	De manière générale, le code est lisible et le style est bon. Il y a quelques problèmes d'indentation et l'alignement des lignes reportées sur plusieurs ligne n'est pas constant et ne respecte les conventions. Il y a également une ligne plus longue que la limite autorisée (87 caractères).
325961	0.50	[L1] 29,144,151; [L2] 46,214,289; [R2]20	En général, le style est clair. Il y a quelques incohérences dans le style d'accolades. Attention, les variables globales ne sont pas autorisées.
326003	2.00		Le code est très bien indenté et structuré. Pour la comparaison avec les lettres majuscules, on privilégiera l'usage de 'A' et 'Z' par exemple, plutôt que la valeur brute en code ASCII (l.118).
326025	1.50	[L1]165-167,253,254 ; [L2]8,42,239	style assez compact ; une ligne vide de temps en temps est utile pour distinguer les déclarations et les blocs de contrôle principaux ; quelques oublis d'indentation et lignes trop longues. globalement le projet semble proche d'une version fonctionnelle.
326030			
326251	2.00		Excellente indentation, rien à redire. Vous avez utilisé une longueur de ligne maximale de 72 alors que elle est de 87 pour ce cours, vous auriez donc pu avoir des lignes plus longues mais cela ne pose pas de problème.

326346	2.00		Le style est presque parfait bravo! L'indentation est constante, comme mentionné avant, il n'est pas nécessaire d'aérer autant votre code. Petit warning également sur des boucles if vides (186, 372) qui sont à éviter à l'avenir, mais pas pénalisées ici, essayez également de garder la fonction main comme première du programme pour plus de lisibilité.
326349	2.00		Code très soigné et très propre. Attention aux lignes 333-334 où l'alignement n'est pas parfait. Il aurait été meilleur d'aligner le "?" et les ":" par exemple.
326369	2.00		Les conventions de programmation sont bien respectées. Bravo !
326417	1.50	[L1]351;[L22]134	Très bon style de code, quelques petites inattentions, mais très bon dans l'ensemble!
326463	1.00	[L1]184,254,362,384 ; [L14]320,410	Le style est bon et la lecture est agréable. Cependant, faites attention à garder la même configurations des "{" tout au long du code, surtout pour les "else". Soyez prudent avec les boucle while(true), une solution avec une condition de sortie est généralement préférée. Essayez de ne pas avoir de code avec trop de if/else/for imbriqué, limitez vous à 3-4 niveau pour que cela reste compréhensible.
326469			
326477	1.00	[L14]101, 202,...; [L22]222-225, 255-256,...	De manière générale, le code est plutôt soigné. Malgré tout, l'alignement des blocs d'instructions et des fonctions est assez peu constant tout au long du programme.
326790	0.50	[L1]17,47,141,235,244,251;[L2]61,97;[R2]54,55,56,208,209	Faire attention aux variables globales et autres conventions ! Ecrire "if(!test)" à la place de "if(test)==false)".
326822	0.00	[L1]99,158,169,204,213,266-267,405 ; [L11]176-186,307-318 ; [L14]259,331,394-396 ; [L2]111,126,168,216,181,185	Plusieurs erreurs d'indentation rendent quelques passages difficiles à lire. Warning L306 est_Majuscule renvoie un bool, pas besoin de le comparer avec false. D'ailleurs ce bloc n'exécute rien, il est donc inutile.
327096	1.50	[L1]282-288,467-470	beau travail ; code très lisible; seulement deux problèmes d'indentation, conseil: remplacez les lignes 503-506 par return (nbT_alpha == contenu);
327283	2.00		Les conventions de style sont tout à fait maîtrisées. Petite remarque : lorsque l'on effectue une longue déclaration, il est préférable d'aligner au début de la parenthèse (ligne 41 du code) plutôt que sur la fin pour faciliter la lisibilité de la déclaration. Aussi, au lieu de vérifier les majuscules une à une, on pourra simplement se servir du code ASCII pour la vérification.
327501	1.00	[R2]12-17,[L14]154-163	Il y a des variables globales à éviter et quelques problèmes de double indentation.
327538	2.00		Le style est très bon ! Il faudrait néanmoins éviter d'appeler une fonction printf, étant donné qu'il existe déjà une fonction avec le même nom dans la bibliothèque standard de C++.
327637	2.00		Le style du code est homogène, concis et agréable à lire. Pour améliorer encore la lisibilité, une ligne vide de temps en temps pourrait être ajoutée afin de mieux pouvoir distinguer les déclarations des blocs de contrôle principaux.
327683	2.00	[L14]256,266	Les conventions sont maîtrisées dans l'ensemble, attention aux doubles indentations cependant.
327694			
327708	2.00		L'indentation est impeccable, le code est agréable à lire et aéré, bravo !

327732	1.50	[L1]66,86,103,301,496-498	Très bon style de code, attention à l'indentation à certains endroits!
327821	1.00	[L11]55-68;[L22]53	L22: Séparation d'un cout en deux seulement si c'est justifié (trop long par exemple), ce qui n'est pas le cas ici Attention aux lignes vides inutiles de partout, il est possible d'en ajouter pour aérer le code mais cela doit être consistant.
327832	0.50	[L14]341-344,[L2]13-18,217,220,...,[L1]76,242	Les conventions de programmation ne sont pas respectées hélas. Pense à contrôler cela la prochaine fois.
328056	1.50	L[1]54,109,157,177	Le code est parfaitement lisible, il faut juste faire attention à la cohérence du style de l'indentation des instructions "if". Sinon c'est super dans l'ensemble !
328073			
328162	2.00		Indentation impeccable, le code est parfaitement lisible et bien commenté. Bravo !
328194	2.00		Très bonne indentation, le code est très clair. Attention, le nom d'un enum doit commencer par une majuscule.
328239	1.00	[L1] 347-348, 350-354, 384-385, 391-393,... ; [L14] 282-284, 300-305, 316-331	Il y a plusieurs styles différents utilisés pour les instructions contrôlées simples et il y a des doubles indentations de certains blocs dans le code. Le style est un peu inhomogène, mais globalement le code reste quand même plutôt bien organisé et compréhensible. Autres commentaires: une ligne vide de temps en temps est utile pour distinguer les déclarations et les blocs de contrôle principaux; pour rendre le code plus compréhensible comparaison avec constantes char 'A' ou 'Z' plutôt que des valeurs brutes de code ASCII.
328245	2.00	L1[61]	Très bon style. Le début du code pourrait être plus aéré avec des espaces. Warnings : espace en trop L61, type coupé L275-276, indentation des else inconsistent (fonction tri L204-228 / L326)
328331	1.50	[L1]52,55,77,195,197-200,212,235,	À part quelques inattentions dans l'indentation, le code est bon.
328354	2.00		Excellent travail ! Attention à bien garder un style d'indentation constant tout au long du fichier, même pour les instructions uniques dans les boucles if, for,... À part ça, le reste est vraiment très bon, n'hésitez pas à ajouter quelques commentaires de temps en temps pour encore améliorer la lisibilité du code.
328390			
328433	0.50	[L1]39-40,84-85,91-92,128,130,...;[L11]255-258;[L2]31,205	Attention à l'indentation qui n'est pas souvent respectée ainsi qu'aux lignes trop longues.
328434	1.50	[L11]15-19	Très bon style de code, attention on petit oubli d'indentation dans la structure.
328436	0.50	[L1]95-97,210 234,262,355;[L14]113-114,126;[L2]32,372	Attention, pleins de petites fautes de styles dans le code, autrement bon style.
328449	1.00	[L1]25-45,53,58,112,114,270-282,319-366;[L2]134,332	Il y a plusieurs erreurs d'inattention concernant le placement des crochets ainsi que l'indentation de la fonction triMot. A part cela, le style est bon.
328515			
328686	0.50	[L11]34-82,96-137;[L14]7-10;[L2]87,91,96,149,171,172,186	Attention aux conventions et plus particulièrement l'indentation ! Ces dommages car des fonctions comme "reviser_doublons" les respectent presque toutes.

328696	2.00	[[L1] 67)	Le style est bon sur le code donnée, attention cependant, il ne peut pas compiler a cause d'un mauvais prototype de fonction et d'une ligne à moitié écrite, Soyez attentif à ça. Nommer un typedef "Vector" est une très mauvaise pratique, car il peut tout à fait être confondu avec le vector simple, cela peut amener à des fautes de frappes difficile à repérer ou à une mauvaise compréhension de la part d'un relecteur.
328879	1.00	[L22]27,39,159,...,[L14]123-142	Les conventions de programmation sont respectées sauf la partie sur les lignes trop longues mises sur plusieurs lignes (L22). Il y a en outre quelques problèmes d'indentation.
329093	2.00		Très bon !
329098			
329110	2.00	[L1]78	Les conventions de programmation sont respectées à une faute de distraction près. Bravo !
329113	2.00		Code très propre et très lisible. Bravo.
329169	1.00	[L1]347:363 [L14]209:211 [L22]143	Attention aux quelques erreurs de style, autrement les convetions de programmations sont comprises.
329286	1.50	[L1]95; [L14] 51-53	De toutes petites erreurs d'indentation mais le code est agréable à lire dans l'ensemble. Le code est très bien commenté, bravo !
329309	1.50	[L2]19,20,21,...;	Bon style en général. N'hésitez pas à définir des alias à l'aide de typedef afin de raccourcir les lignes dans l'ensemble du code, ceci vous aurait permis d'éviter les nombreuses déclarations de fonctions trop longues.
329328	1.50	[L14]124, 194, 195, 196, 272	Les conventions de style sont respectées, attention cependant aux doubles indentations par endroits.
329464	0.00		fichier sans code
329505	1.00	[L1]116, 284, 321 [L2]60, 73	Bonne indentation assez cohérente malgré quelques oublis, revenez à la ligne avant d'attendre la limite de votre écran en alignant proprement avec le début de la liste de paramètres (ou de la condition le cas échéant)
329648	1.00	[L1]94-95, 103; [L2]65, 66, ...	Dans l'ensemble, code plutôt bien lisible. Vous avez cependant oublié la limite de 87 caractères par ligne. De plus, une condition if contenant deux instructions doit s'écrire sur plusieurs lignes et le style if(){...} sur une seule ligne n'est pas consistante avec les conventions de programmation.
329681	1.50	[L1]105,138-146,321	code globalement agréable à lire. quelques indentations incorrectes. remplacer 416-421 par return (a<=b);
329722	0.50	L[1]142-144,333,334,398; L[14]290; L[22]39-53,331,332	Le code est globalement agréable à lire, mais il faut faire attention aux indetentations qui font malheureusement perdre quelques points. En revanche le code est super bien commenté et aéré ce qui augmente grandement la lisibilité !
329723	1.00	L[1]60,88,82,97-101; L[14]45-48,82-89,113-118	Le code est globalement agréable à lire, mais il faut faire attention à l'indentation qui n'est souvent pas bonne.
329756	2.00		Bon style !
329782	2.00		Le travail global est vraiment excellent voire parfait. Le style est très propre et agréable à suivre. Bravo c'est un vrai plaisir de lire un projet pareil.
329793	2.00		Toutes les conventions sont bien respectées. Bravo.
329874			

329887	2.00		Les conventions sont très bien respectées ! Bon style en général ! Noter qu'il est possible d'enlever les accolades d'un branchement si celle-ci ne possède qu'une seule instruction.
329904	0.50	[L1]48-50,242,288-289;[L14]13-17;[R2]35	Bon style ! Attention, les points virgules après les accolades sont inutiles (e.g ligne 66). Faire aussi attention aux conventions sur l'indentation et sur les variables globales.
329910	1.50	L[14]76,128	Le code est très agréable à lire et bien commenté, l'indentation est très bien respecté dans sa globalité, il faut seulement faire attention à ne pas faire de double indentation.
329920	2.00		Indentation parfaite, très cohérente, n'hésitez pas à laisser une instruction unique sur la même ligne que le if notamment pour tri_dico qui pourrait être raccourcie
329942	0.50	[L1]85, 355, 358, 476 [L14]235, 264, 265, 266 [L2]39,41, 388	Attention, beaucoup de fautes d'indentation et quelques lignes trop longues, autrement le style est bien.
330043	2.00		Les conventions sont très bien respectées ! Bon style, il est possible d'enlever les accolades d'un branchement si celle-ci ne possède qu'une seule instruction.
330070	2.00		Les variables et les paramètres ne devraient pas commencer par une lettre majuscule. Néanmoins le style du code est excellent.
330101	1.50	[L2]28,45,57,295,522	Bon style de code, attention à respecter les 87 caractères maximum par ligne. Les constantes d'erreurs auraient pu être déclarées au sommet du code, car elle sont constantes.
330128			
330146			
330186	0.50	L[2]215,254; L[22]213-215 R[2]128	Le code est agréable à lire dans son ensemble et il est très bien commenté. Il faut faire attention en revanche aux lignes trop longues, à l'indentation des grosses conditions dans les boucles "while", et il ne faut pas déclarer de variable globale.
330541			
330583	0.50	[L11]85,110 ; [L14]67,76 ; [L2]45,210,283	Fais attention au style, ton code est bon, c'est dommage de perdre des points comme ça.
330629			
330640	2.00		
330667	2.00		Excellente indentation, style consistant.
330686	1.50	[L1]29,30,48	indentations variables (1, 2 ou 4 caractères), warning: les commentaires ne doivent pas dépasser la limite de 87 caractères. Ligne16: un nom 100% en majuscule est seulement pour des symboles créés avec #define.
330762	1.50	[L14]285,365;	Très bon style. Attention aux petites inatentions sur la double indentation, et essayez d'aérer un peu plus votre code à l'aide d'espaces et de retours à la ligne. Favorisez également l'utilisation de "unsigned" à "size_t", ce dernier étant surtout là pour le parcours de tableaux.

330833	1.00	[L1]229,245(Brace style not consistent);[L2]202,234,264,267	Bon style en général mais attention au conventions : le style des accolades doit être le même dans tout le code; le nombre max de caractère par ligne : 87 Aussi les noms de variables et fonctions sont trop long ce qui oblige à souvent couper des lignes en plusieurs plus courtes, cela rend l'ensemble moins lisible. Exemple : <code>get_and_print_all_message_anagrams, sort_dico_alphabetically.</code>
331471	1.50	[L22]112,127,263,268	Très bon travail dans l'ensemble, attention toutefois à garder un style constant et bien défini lorsque vous avez des instructions sur plus d'une ligne. Essayez aussi de mettre des accolades quand bien même il n'ya qu'une seule instruction suivant la boucle, car on se perd un peu dans les enchaînements de for et de if.
332230	1.50	[L11]14-17	Très bon style de code, une petite inattention dans le bloc de la structure, mais autrement super travail!
339389	1.50	[L1]128, 142, 153, ...;	De manière générale, le code est plutôt soigné. Malgré tout, le style des accolades pour les instructions n'est pas cohérent tout au long du programme.
339407	0.50	[L1] 54,369 ; [L11] 41-72,89 ; [L22] 20,36,61,195,201,208	Dans l'ensemble le style est ok, mais n'oubliez pas que le main doit être aussi indenté, comme une fonction "normale". Soyez attentif à l'indentation des blocs de contrôle, ne laissez pas traîner d'espace inutile (l.369). Quant vous devez écrire une instruction sur plusieurs lignes, prenez le temps d'aligner les lignes suivantes sur la première pour améliorer la lisibilité. Le cout de la ligne 61 aurait pu être séparé en plusieurs petit cout, ou encore mieux, vous auriez pu stocker le string sous forme d'un const string placé au début du programme. Mais dans l'ensemble c'est bien !
339421	1.50	[L2]74, 124, 158	On remarque les soins apportés à la présentation du code, mais il n'est pas nécessaire de commenter chaque lignes. Attention, il y a beaucoup trop d'espaces inutiles Commenter la fonction "calcul_nbD" avec le commentaire "Calcul du nbD" n'a pas vraiment d'intérêt.
339442	1.50	[L2]128,135,287,290,313,355	Très bon style de code, attention aux lignes faisant plus de 87 caractères!
339450	2.00		Très bon style en général ! Warning ligne 169 La séparation en plusieurs lignes de la condition a bien été faite, cependant cela aurait été plus lisible en décalant plus vers la droite pour mieux aligner les conditions.
339452	0.50	[L14]55-62;[L2]51,68,87,90;[L22]13,17,24,77,305,360	Attention aux conventions qui ne sont pas respectées ! Le code est parfois trop ou pas assez indenté. La séparation d'une ligne en plusieurs est possible si celle-ci dépasse 87 caractèresn, autrement ce n'est pas nécessaire de la séparer.
339458	2.00		Bonne indentation, n'hésitez pas à espacer un peu plus votre code et vos expressions mathématiques avec des espaces pour les rendre plus lisibles
339621	2.00	[L2]55	Les conventions de programmation sont respectées. Bravo !
339646	2.00		Très bon style de code, à continuer pour la suite!
339649	1.50	[L14]109-110, 133-155, 186-189, 220-225	On remarque les soins apportés à la présentation du code, cela dit l'indentation n'est pas toujours correcte. Le nom des fonctions et des variables est beaucoup trop long, cela rend les expressions très verbeuses et oblige à séparer les expressions sur plusieurs lignes. Par ailleurs, faire attentionà l'utilisation souvent abusive de la séparation sur plusieurs lignes d'une expression (e.g : ligne 594).

339655	2.00		Aucun problème d'indentation, vous pourriez revenir à la ligne encore plus souvent pour les conditions composées de plusieurs AND enchainées, le code est très agréable à lire
339712	1.50	[L1]14, 37	Excellente indentation malgré un petit oubli dans le main
339713			
339719	0.50	[L1]117, 130, 156, [L14]127:131, 140:146, 155, 156, 17:20, 206, 207 ... [L2]10,12, 20, 65, 72, ...	Attention aux conventions de programmation et particulièrement à l'indentation.
339831	0.00	[L1]95,125 ; [L11] 47-99 ; [L14]60-66,94 ; [L2]23,45,62	Le style n'est pas consistant sur tout le code, l'indentation est mauvaise, les accolades ne sont pas alignées. Tout cela rend le code difficile à lire
339870			
339943	1.00	[L1]18;[L2]42,45	Le style des accolades ligne 18 n'est pas consistant avec le reste du code. Travail insuffisant
339959	1.50	[L1] 172, 199, 229	Les conventions de styles sont respectées, attention à l'indentation dans les différents blocs (if,else,for, while, ...)
340062	0.50	[L1]43,56 ; [L11]43,56,58 ; [R2]12-14	Fais attention à bien indenter ton code à l'intérieur de tes fonctions et des boucle while. Essaie de ne pas utiliser de variables globales.
340374	0.00	L[1]122,139; L[11]219-415; L[14]101,125,156,240,243; L[2]163,166,287,325,388; R[2]21-30	Le code est vraiment bien commenté mais l'indentation n'est souvent pas bonne. Certaines lignes sont trop longues et il faut faire attention à ne pas déclarer de variable globale.
340423	2.00		Attention à la limite de caractères qui a été dépassée une fois (ligne 393). Sinon les conventions de styles ont été respectées.
340497	2.00		Les conventions sont très bien respectées ! Lorsqu'un branchement ne possède qu'une seule instruction, il est possible de se passer des accolades.
340645	2.00		Excellente indentation, rien à redire, les commentaires sont très pertinents et appréciés
340769	1.50	[L22]13,15,18,19,26-28,...	Le choix des tabulations de deux espaces est curieux mais autorisé. En revanche, les cas de fonctions trop longues sont mal traitées.
340803	1.00	[L1]288, 290, 304 [L14]354	Bonne indentation malgré un manque de consistance pour les instruction uniques suivant un if, les structures de little_sort et merge pourraient être améliorées.
340814	1.00	[L1]516, 522, 556, 570 [L22]534, 539	Bonne indentation malgré quelques manques de consistance sur des if et des for à instruction unique. N'hésitez pas à vous aligner avec le début de la liste de paramètres lorsque vous revenez à la ligne pour écrire la fin de la liste de
340840	2.00		Le style est impeccable et le code est facile à lire.
340863	1.00	[L2] 38, 39, 40, 191 [R2]7, 9, 10, 11	Attention aux variables globales ainsi qu'aux lignes trop longues. Sinon le code est propre et le style respecté.
340864	1.50	[L14]184:193	Hormis une double indentation, le style est très bien

340932	1.00	[L1]66,84,237,371,...; [L11]250;	Bon style dans l'ensemble, attention à garder une indentation constante tout au long du programme même pour les instructions uniques. Le code est aéré et très bien commenté, continuez ces bonnes pratiques!
340933	1.00	[L1]72, 73, 167 [L14]80, 83	Dans l'ensemble le style est bon malgré quelques petites erreurs.
340938			
340944	1.50	[L1]231 [L14]58:84, 108:116, 217, 241, 259	Dans l'ensemble, les conventions de programmation sont respectées, attention cependant aux doubles indentations.
340949	0.50	[L14]77, 78,79, 101:111, 315, 316, ... [L2]128, 324 [R2]7	L'utilisation d'un mode debug est une bonne pratique, cependant utilisez constexpr ou define pour cette variable globale. Attention également aux doubles indentations et aux lignes trop longues.
340952	1.00	[L1]254, 353, 372, 373, 377, 413, 568, 743, 790, 816 [L22]35, 39, 51, 167, 443, 511, 561, 676, 703, 808	Les conventions de styles sont globalement respectées. Attention à la convention utilisée pour les accolades qui doit être constante (ici, on a un mélange de 2 conventions valides) ainsi qu'à l'alignement, pour la lisibilité, il est préférable d'aligner avec le début de la première variable d'une parenthèse, par exemple.
340991	2.00		Conventions de style impeccables. Alignements très maîtrisés et facilitent grandement la lecture du code.
340992	2.00		Conventions de style parfaitement maîtrisées.
340997	2.00		Conventions de style parfaitement maîtrisées.
341008	2.00		Excellente maîtrise des conventions de style. Attention à l'indentation des boucles if, for, while (une seule erreur commise à la ligne 418).
341015	0.50	[L1]59,71; [L22]27; [L14]218- 223	On nomme en général les variables de manière parlante, pour mieux comprendre leur but. De plus, les commentaires qui apportent de l'information sont à privilégier. Attention aussi à garder un style d'accolades consistant.
341021	2.00		Le code est très bien structuré et clair.
341042	1.00	[L1]293,373; [L14]342-354	L'indentation et le style sont généralement très bons. Attention à rester cohérent dans la convention utilisée pour les noms composés.
341045	2.00		Le code est bien indenté et le style est cohérent. Attention à la convention sur les noms composés (E121).
341057			
341069	1.50	[L1]6,32,110	Bonne indentation, quelques incohérences sur les accolades. Attention, on privilégie plutôt la comparaison avec les caractères 'A' et 'Z' plutôt que les valeurs en code ASCII (l.79,85).
341115	0.50	[L1]88,122,124,131; [L11]35- 43; [L14] 94,100,101	Le code est généralement bien indenté et structuré, sauf pour une fonction. Le nom des paramètres de certaines fonctions pourrait être plus clair. Pour la comparaison avec les lettres majuscules, on privilégiera l'usage de 'A' et 'Z' par exemple, plutôt que la valeur brute en code ASCII (l.100).
341127	2.00		Les conventions de programmation sont bien respectées. Bravo. Pense peut-être à d'avantage mettre de retour à la ligne pour "aérer" ton code.

341138	1.50	[L2]119,194	Le code est très bien présenté et il respecte les conventions de programmation sauf sur la contrainte des lignes trop longues. En effet, certains commentaires dépassent hélas la limite autorisée.
341148	1.00	[L1] 260, 264 [L11]245	L'indentation n'est pas toujours idéale, n'hésitez pas non plus à aérer un peu votre code, la structure en elle-même est bien répartie
341178	2.00		Il serait meilleur d'utiliser true et false à la place de 1 et 0 et il faudrait éviter de mettre un while(true). A part cela le style du code est très bon !
341181	2.00		Ajouter des espaces et des lignes vides rendrait le code plus facilement lisible, néanmoins le style est impeccable !
341189	1.50	[L14]168-170	Le style est très bon ! Ajouter des lignes vides pourrait aider à rendre le code encore plus lisible.
341215	2.00		Le code est très bien documenté et le style est très bon !
341237	2.00		Les conventions de programmation sont minutieusement respectées. Bravo !
341270	1.00	[L1]50,78,83,174,...,[L2]31-33	Les conventions de programmation sont peu respectées en ce qui concerne les indentations et les alignements du code.
341273	1.50	[L1]146,264,318,...;	Excellent travail en général. Petit bémol sur l'indentation d'instructions contrôlées simples où elle n'est pas constante tout au long du programme, à part cela, le style est presque parfait! Essayez peut-être de raccourcir un peu le nom de certaines fonctions (telles nombre_de_repetitions) qui peuvent les rendre plus courtes sans nuire à la clarté de leur action.
341284	1.50	[L14]561;	Excellent style! Juste une petite inattention de double indentation mais à part ça le code est très aéré sans trop l'être et est très agréable à lire!
341354	2.00		Excellente indentation très cohérente, le code est agréable à lire.
341363	2.00		L'indentation est parfaite, hésitez pas à commenter un peu plus. Le code est agréable à lire dans l'ensemble !
341372	1.00	[L1]49,133,325,373; [L2]189,373	Le code est agréable à lire. Attention aux lignes trop grandes et à la cohérence du style de l'indentation des instructions "if".
341400	1.50	[L11]L323-324	Très bon style, mais quelques erreurs d'inattention : oubli d'un retour à la ligne L160, indentation L200 inconsistente. Faire attention pendant la relecture en général.
341439	1.50	[L1]73,88,90,91	Le code est bien indenté et agréable à lire. Dommage que le style d'accolades ne soit pas cohérent sur l'ensemble du
341450	2.00		Très bon style ! Essaie juste de corriger les warnings de ton compilateur, aussi tu as tes variables non utilisées comme à la ligne 643.
341451	1.50	[L11]74,170,282	Petites erreurs d'indentation, mais très bon style en général. Petite inconsistance L430.
341489	2.00		Excellente indentation, style consistant. Warning : uint n'est pas un type standard, préférer size_t. Warning 2 : le main devrait être après le prototypage.
341551	2.00		Bravo, le style est très bon et très aéré, rendant le tout très agréable à lire. L'indentation est quasi parfaite et les noms des variables/fonctions très bien choisis. Essayez peut-être d'espacer un peu plus les prototypes des fonctions, le nom de la structure n'est pas forcément parfait mais ne pose pas de problème non plus. N'hésitez pas non plus à ajouter quelques commentaires de temps en temps afin d'augmenter encore la lisibilité du programme. Globalement c'est du très bon travail!
341622	1.50	[L1]17 ; [L2]191,200	Excellente indentation, style consistant. Attention, les commentaires comptent aussi pour le nombre max de caractères sur une ligne.
341668	2.00		Indentation impeccable, le code est parfaitement lisible et bien commenté. Bravo !

341670	1.50	[L1]92,104,130,147	Le code est parfaitement lisible et bien commenté, il faut juste faire attention à la cohérence du style de l'indentation des instructions "if". Sinon c'est super dans l'ensemble !
341687	1.00	[L14]339-372 ; [L22]36,40,364,392,434	Conseil : L190 operateur_inf_mot renvoie un bool, pas besoin de le comparer avec false. Conseil 2 : vous auriez eu moins d'erreurs de line wrapping si vous utilisez des typedefs et abréviations comme dico pour dictionnaire. Warning : faire attention aux conventions d'organisation, ex. : le main devrait être au début du code. Warning 2 : l'alignement des paramètres avec une indentation induit des erreurs : vous avez fait une double indentation pour la fonction rech_anagramme à cause de ça.
341739	2.00		Excellente indentation, bien qu'agréable pour la relecture, vous n'avez pas besoin de sauter des lignes partout, cela vous permettra d'avoir un code plus compact et plus rapide à naviguer
341743	1.00	[L2]10, 22, 24, 31, ... [R2]10	Aucun retour à la ligne dans le code, revoyez les conventions de programmation à ce sujet. La variable dicobrut n'est ni constexpr ni un define et n'est donc pas accepté pour ce projet
341899	1.50	[L14]16-21	Le style du code est très bon !
341931	1.50	[L22]143	Excellente indentation, style consistant. Petite erreur d'alignement L143.
341932			
341940	2.00		Le code est plutôt homogène et bien indenté (juste une indentation de trop (lignes 194-197), et 1-2 fautes (lignes 11, 391)). Pour améliorer le code la comparaison avec constantes char 'A' ou 'Z' plutôt que des valeurs brutes de code ASCII est préférable.
341951	1.50	[L1]124, 228, 308	Dans l'ensemble très bon style. Quelques problèmes d'indentation et d'accolades vous enlèvent quelques points.
341953	2.00		Bon style. Tu peux l'alléger, par exemple l.347, au lieu de faire if(a<=b)return true ; else return false ; , fais juste return a <= b.
341970	2.00		L'entier du code est parfait d'un point de vue stylistique, bravo! J'ajouterai que faire ce projet en aussi peu de lignes est vraiment très fort! C'est de l'excellent travail
341975	1.50	[L1]106,107	Bon style, fais attention à avoir le même style partout pour les if avec une seule ligne.
342052	1.00	[L1]390,431 ; [L14]379,381	Bon style, attention cependant aux indentation en trop ! Ne laissez pas trop de ligne vide à la fin, elle peuvent contenir des caractères invisibles et être source d'erreur de compilation. N'hésitez pas à combiner les conditions avec "and" au lieu de faire des successifs quand cela est possible. Pour raccourcir les conditions, vous pouvez utiliser le "!" pour faire le not et remplacer if((cond) != true) par if(!(cond)).
342054	2.00		De manière générale, très bon style. Attention cependant à être consistant dans la façon de mettre les accolades pour les conditions if d'une seule ligne.
342060	0.50	[L1]69,71,78,...; [L14]190,353,457; [L2]7,41,45,...;	Les points obtenus ne sont pas vraiment à l'image du style qui est bon mais il y a plusieurs erreurs d'inattention. Faites bien attention à aligner l'accolade fermante avec le début de l'instruction, en réduisant la taille des noms des fonctions vous auriez pu éviter d'en avoir des trop grandes. Globalement agréable à lire mais aurait demandé un peu de "nettoyage" du point de vue des conventions.
342063	1.00	[L1]150,195,219,237,291 ; [L14] 65	Style OK, attention à ne pas mélanger les espaces et les tabulations pour l'indentation, choisissez-en un et gardez-le pour tout le projet. Attention aux return, ils se placent après un retour à la ligne et non directement après le }. Les lignes vides sont bien pour la lisibilité, mais attention de ne pas en abuser (cf. anagramme_maker).

342154	0.50	[L1] 106-108, 271-273, 276-277; [L14] 90, 95, 155, 266-281,; [L22] 250-251, 402-	Le code n'est pas toujours correctement indenté. Par contre, il est bien organisé et on s'y retrouve bien.
342200	1.50	[L14]281, 282	Code très lisible et très propre. Il y a un accolade mal positionnée à la ligne 66 (non sanctionné) et une indentation en trop pour les lignes 281 et 282.
342204	1.50	[L2] 225, 273, 294,...	L'indentation du code est homogène et rend le code facile à lire. Cependant, il faut faire attention aux wrapping lines.
342215	2.00		Bon style. Tu devrais essayer de corriger les warnings du compilateur ; c est possible d'alléger ton code, par exemple la fonction rangement, declare plutot ton vecteur de tabl avec une taille initale (dico.size()) pour éviter de faire des push_back(tabl()).
342216	2.00		Bon style, le code est bien lisible.
342224	0.50	[L1]212,284-285;[L14]176,352,354,359,434-435;[L22]222-225,239-	Il y a de multiples problèmes d'indentation et d'alignement, néanmoins le reste du style est très bon.
342283	1.00	[L1]295-344,304-322,309-316,324-327;[L11]194-366	Par convention les définitions de structures, de types et de constantes devraient être placés au début du code. Il y a quelques erreurs d'indentation dans le main mais le reste du code a un bon style !
342391	1.00	[L1]65,208 ; [L2]101,277	Fais attention au style, c'est dommage de perdre des points comme ça pour un si bon code. N'indente pas tout ton code !
342575			
342745	1.00	[L14]54-60, 196; [L22]72	Code très lisible et très propre. Dommage que quelques lignes soient trop indentées et que l'alignement des lignes se reportant sur plusieurs lignes pose problème à certains endroits du code.
342800	1.00	[L22]17	Le style général est bon, les conventions sont bien respectée dans l'ensemble. Il n'est pas nécessaire, voire faux de mettre des ";" après la déclaration d'une fonction, faites attention ! Autre point, c'est un mauvais style de mettre des méthodes dans une structure, c'est une chose à réserver aux classes. Cela sera même interdit lors du prochain semestre. Les structures doivent rester un container pour mettre ensemble des types de données, sinon l'utilisation des classes est favorisée. Vous perdez ici 0.5pt. Autre recommandation, pour la condition de la boucle for l.203, ne mettez pas un appel de fonction ici. Si le nombre de tour de boucle ne peut pas être connu à l'avance, privilégiez une while, et sinon, utilisez une variable, sinon la fonction sera executé à chaque check de la condition. Soyez constant dans votre style de if avec une seule ligne, choisissez en un et gardez le.
343266	1.50	[L2] 36, 264, 285	Code très agréable à lire, bien aéré et bien indenté. Attention aux wrapping lines. Améliorations: Beaucoup de if(condition) return true; else return false; -> dans ce cas-là on peut directement retourner la condition // aussi pour return pas besoin des parenthèses.
343725	2.00		Le style du code est homogène, aéré et très agréable à lire.
343736	1.00	[L1] 99-100, 101-102, 238-239; [L14] 177-179, 224-228, 249-253	Il y a plusieurs styles différents utilisés pour les instructions contrôlées simples et de plus elles ne sont pas toujours dans les variantes acceptées. Il y a également des doubles indentations dans certains blocs dans le code. Le code est très bien organisé et espacé entre les fonctions (même si espacements pas toujours homogènes). Par contre dans les fonctions, le style est assez compact il pourrait y avoir une ligne vide de temps en temps pour distinguer les déclarations et les blocs de contrôle principaux.

343876	2.00		Excellente maîtrise des conventions de style. Petite remarque, au lieu d'imbriquer plusieurs boucles if les unes dans les autres, il est préférable de mettre les différentes conditions dans une seule boucle ou d'utiliser un switch statement.
344139	1.50	[L2]43,210,212,228,273	Les conventions sont très bien respectées ! Bon style ! Attention à la convention sur le nombre max de caractères par ligne.
344193	1.50	[L14]270-276	Ton style d'accolade rend le code très difficile à lire, je te conseil vraiment de passer tes accolades fermantes à la ligne et mettre une accolade maximum par ligne (pas 3 !). Cela rendrait aussi l'indentation plus facile à faire et à vérifier.
344225			
344276	1.50	[R2]27	Il ne devrait pas avoir de variable globale et les différents mots dans les noms de fonctions pourraient être séparés par un espace souligné (_), néanmoins le style du code est très bon.
344310			
344312	2.00	[L1]99	Bon style dans l'ensemble, attention aux instructions de contrôle sur une seule ligne qui peuvent devenir lourde lors de la relecture du code.
344350			
344385	2.00		Excellente indentation, style consistant.
344399	2.00		Les conventions sont très bien respectées ! Dans la fonction "swap" il est possible dans ce cas de juste écrire "a_echanger[p] = a_echanger[p-1]" pour copier les valeurs des variables
344405	2.00		Les conventions sont très bien respectées ! Attention aux lignes vides qui sont ajoutées aléatoirement (des fois il y en a, des fois pas) (lignes 30, 32, 47, ...) qui rendent le code trop éparpillé.
344415	1.00	[L14]118:126, 141:147, 157, ... [L2]176, 204	Il faut faire attention aux lignes de codes qui sont indentées de 5 espaces au lieu de 4 ainsi qu'à la double indentation dans la fonction tri_nbT. Attention également aux lignes trop longues.
344463	1.50	[L14]100, 102, 36:39, 47:51	Les conventions de programmation sont parfaitement respectées à part quelques oublis d'indentation.
344514	2.00		Excellente indentation, style consistant.
344516	1.50	[L2]217 ; [L22]134,146	Excellente indentation, style consistant. Conseil : Utiliser le type size_t pour ne pas avoir de warning de comparaison entre des entiers signés et non-signés.
344526	1.50	[L2] 23, 48, 602	Le style du code est homogène, aéré et très agréable à lire. Attention cependant aux indentations il y a 2 espaces pour le bloc des fonctions mais sinon un tab pour les autres types de bloc et ceci peut vite devenir difficile à gérer. Il est plus facile de mettre un tab pour tous les blocs. Attention également aux wrapping lines pour les commentaires.
344528	1.50	[L1]175, 179, 183, 189, 188	Code très soigné et style très propre. Simple problème d'accolades fermantes, dont le style n'est pas constant tout au long du programme. Il manque également une indentation à la ligne 188. Dans la même fonction, il y a une indentation de trop. Elle n'est pas pénalisée car très reliée aux autres problèmes.
344538	2.00		Excellente indentation, style consistant. L120-123 et L147-150 sont un peu lourds, les conditions peuvent être mis sur plus de lignes.
344544	1.00	[L1] 16-25, 34, 42, 74-77,...; [L2] 105, 109, 115, ...	Le code n'est pas toujours correctement indenté et il y a des wrapping lines. Il y a également beaucoup de valeurs hard-codées qu'en tant que lecteur on peine à comprendre.

344700	2.00		Les conventions de programmation sont respectées. Bravo !
344708			
344732	2.00	[L1] 13-17	Le style du code est homogène, aéré et très agréable à lire.
344736	1.50	[L14]292	Bon code dans l'ensemble. Tu pourrais faciliter le tri avec une fonction de comparaison de Mot1 et une fonction swap d'éléments dans un vec par exemple.
344793	2.00		Plusieurs fonctions ont trop de lignes vides, ce qui les rend très longues. A part cela le style est très bon !
344804	2.00		Excellente indentation, style consistant. Warning : Faire attention aux conventions d'organisation, ex. : le main devrait être au début du code, il faut un prototypage des fonctions.
344822	2.00		Excellente maîtrise des conventions de style. Petite remarque : il est préférable d'utiliser un "switch statement" plutôt qu'un grand nombre de boucle "if" se terminant par un "break".
345007	1.50	[L2]227,232,234	Le code est très bien documenté et séparé. Le style est très bon.
345020	2.00		Excellente indentation très consistante, les retours à la ligne sont irréguliers mais ne constituent pas un problème en soi
345219	1.50	[L14]585	Le code est bien indenté et structuré, et les commentaires sont pertinents. Pour la comparaison avec des caractères, on privilégie l'utilisation de 'A' et 'Z', par exemple, plutôt que la valeur en code ASCII (l.138,169). Attention, le code contient un bloc conditionnel vide (l.613).
345259	0.50	[L1]264; [L11]123,186,211; [L14]124,363-366; [L2]20,61,69,249,270,333,387	Code globalement agréable à lire, mais quelques mauvaises indentations et certaines lignes qui dépassent 87 caractères qui font malheureusement perdre quelques points. N'hésitez pas à écrire "using namespace std" au début du code pour ne pas avoir besoin de mettre std:: à chaque fois.
345261	0.50	[L1]209,214,325,385 ; [L11]81, 109 ; [L22] 15,17,33,35,37	Le style est loin d'être mauvais, mais beaucoup de petites erreurs d'indentation compliquent la lecture et rendent le programme difficile à suivre. Quant vous revenez à la ligne à cause d'une instruction trop longue, prenez garde à bien aligner la ligne avec quelque chose dans la précédente pour faciliter la lecture. Par exemple pour les if avec des conditions multiples, aligner les conditions entre elle, ou les and entre eux. Mais il y a un bon fond, courage !
345286	2.00		Indentation parfaite, rien à redire
345290	2.00		Le style est bon dans son ensemble, à part les quelques erreurs d'indentation non constante au travers du programme. Les commentaires sont vraiment très bons et aident à la compréhension du fonctionnement du programme, une excellente pratique à continuer à mettre en oeuvre! Le choix de créer une structure au début du code aurait pu simplifier toute la suite.
345298	1.50	[L1]391 [L14]176:200, 369:375, 382:384, ...	Beaucoup de double indentations, autrement le style est bon.

345308	0.00	[L1]68, 105, 118, 175, 219, [L11]70, 191 [L14]246, 248, 263, 305, [L2]37, 59, 61, 65, 128, 139, 148, 212, 219, 236, 239, 291, 308, 319	Il faut revoir les conventions de style afin de favoriser un code lisible et agréable à lire.
345310	1.50	[L2]107, 120, ..	Mise à part que le nombre de 87 caractères par ligne n'a pas été respecté, le style est bon et le code très propre. Rien à redire
345337	1.00	[L1]18,32,36,47,55- 58,...;[L2]101,113,152	Bon style, attention à l'indentation des blocs ainsi qu'aux lignes trop longues.
345346	1.00	[L2]8,19,35 ; [R2]9	L'indentation est bien faite. Warning 1 : les conventions d'organisation ne sont pas respectés. Exemples : main en bas du code, pas de prototype de fonction. Warning 2 : constantes globales. Pour la constante capital_letters, qui pourrait être légitime, utiliser constexpr, enum ou define pour bien indiquer que c'est une constante. Mais vous pouvez directement comparer les lettres, qui sont convertis en nombres selon le code ASCII. Warning 3 : faire attention à bien enlever les stubs (L407).
345352	1.50	[L1]216,221,227,232,234,238,.. ..	Penser à déclarer les prototypes des fonctions au sommet du fichier ainsi que les structures plutôt que de les insérer au milieu du code.
345393	2.00		Indentation impeccable, le code est bien commenté. Il faut en revanche ne pas hésiter à sauter des lignes entre chaque fonction pour améliorer la lisibilité. Il faut également faire attention à ne pas laisser des bouts de code en commentaire dans la version finale. Attention aussi aux noms des fonctions : par exemple au lieu de nomer la fonction "detectiondoublon", les conventions veulent qu'on la nome "detectionDoublon" ou "detection_doublon". Sinon c'est super dans l'ensemble !
345420	2.00		La fin de la fonction tri_dictionnaire est écrite d'une façon très peu recommandée, mais à part cela très bon style !
345431	1.50	[L22]78, 118, 327	Les conventions sont respectée dans l'ensemble. Pensez toutefois à aligner les lignes quand une instruction ne rentre pas sur une ligne
345442	2.00		Bon. Juste ton style des accolades fermantes n'est pas très classique, en générale on les met au même niveau que celles ouvrantes.
345448	2.00		Le style du code est homogène, concis et agréable à lire.
345459	0.00	[L1]59,191; [L11]293-297; [L2]76,86; [L22]240,378	Les conventions sont majoritairement respectées, mais il y a beaucoup de fautes d'indentation isolées qui s'accumulent. Plusieurs lignes ne respectent pas la limite de 87 caractères. Attention, une variable avec un nom tout en majuscules ne respecte pas les conventions (E12). De plus, donner à une variable le nom d'un type peut rendre le code confus, voir mener à des erreurs. Il y a également beaucoup de commentaires, qui n'apportent pas tous de l'information utile.
345473	1.00	[L2]29,57,12,...,[L1]38,52,90,1 05,110,...	Il y a de nombreux problèmes au niveau de la présentation du code. Pense à relire le document sur les conventions de programmation.

345489	1.50	[L1]58,74,...427;	C'est du bon travail dans l'ensemble, essayez toutefois de garder le même style d'indentation tout au long du fichier et ce pour n'importe quel type d'instruction (fonction, if, for,...), cela aide grandement à la lisibilité du code. Je recommande d'ajouter quelques brefs commentaires de temps en temps également.
345491	2.00		Le style est parfait. Pas grand chose à redire. Faites simplement attention à ce que les conditions if d'une seule ligne conservent le même style tout au long du programme (petit problème non pénalisé ligne 423).
345522	2.00	[L2]44, 68, 70, 121, 131, 153, 174, 232, 250, 252, 263, 276	Les conventions de styles sont respectées, attention au dépassement de la limite de 87 caractères.
345577	2.00		L'indentation est parfaite, hésitez pas à commenter un peu plus. Le code est agréable à lire dans l'ensemble !
345618	2.00		Le style est très bon ! Ajouter des lignes vides pourrait aider à rendre le code encore plus lisible.
345671	1.00	[L1]69,81,96,134,187-188 ; [L2]28,159,164,169,217,225	Style ok. Attention à l'indentation et au dépassement des 87 caractères par ligne.
345710			
345712	2.00		joli style, attention à la petite erreur d'indentation à la ligne 133. Faites aussi attention à ne pas avoir de fonction trop "profonde", avec trop de for/if/else/while imbriqué.
345716	0.50	[L1]145-146,211,291;[L14]12-15;[L22]45,258,271	Attention aux petites inattentions s'accumulent malheureusement. Penser à utiliser des tabulations plutôt que des espaces pour l'indentation rend le tout plus simple.
345765	1.50	[L2]159,163	Les conventions sont respectées sauf dans une condition bien trop longue.
345770	1.50	[L1]288,307	Bon style, fais attention de bien mettre tes accolades fermantes toujours au même niveau partout.
345772	1.00	[L14]39,72,89,101,114,147,159,... ; [L1] 107,173	Dans l'ensemble le style est bon, faites cependant attention à ne pas faire de double indentation, vous le faites tout au long du programme, notamment lors des boucles for (205) ou dans certaine fonction (main et affiche_dico). N'hésitez pas à demander au semestre prochain à un/e assistant/e de vous montrer comment les identifier correctement.
345773			
345785	1.50	[L22]119-120, 124-125,...	De manière générale, très bon style. Attention cependant à être consistant dans la façon de mettre les accolades pour les conditions if d'une seule ligne. Les longues instructions de plusieurs lignes pour les conditions doivent être alignées avec le début de l'expression évaluée.
345853	0.50	[L1]12,52,62; [L14]21-30,32; [L11]47-60	Pour plus de lisibilité, l'utilisation de prototypes permet de déclarer les fonctions après le main. Attention aux accolades et à l'indentation. Pour une struct, le nom doit commencer par une majuscule.
345863	2.00		Joli code, bravo ! Essayer de mieux séparer votre code en fonction pour éviter d'avoir trop de for et de if/else imbriqué qui rendent le code difficilement lisible (notamment la fonction tri_dico). Si ce n'est pas possible, essayer au moins d'aérer les fonction compliquées en séparant les fonctionnalités
345866	2.00		Code très soigné et style très propre. Vraiment rien à redire.
345936	1.00	[L1]220;[L11]15-18;[L2]178;[L22]296,299	Attention aux petites inattentions s'accumulent malheureusement. Penser à utiliser des tabulations plutôt que des espaces pour l'indentation rend le tout plus simple. Certaines lignes vides pourraient être supprimées (comme par exemple dans un bloc de code avec une seule ligne, rajouter une ligne vide au début rend le code plus "lourd". Autrement
346039	1.00	[L1]80,115,133,156-157,168,261-263,...;[L14]86-92,116-127,203-208	Attention aux indentations! Bon style autrement dans l'ensemble.

346080	0.50	[L1]333,352,373,392,432,451;[L11]12-15;[L14]333,352,373,392,432,451	A plusieurs endroits les crochets ne sont pas indentés correctement et pas mis sur des lignes différentes. Le style du reste du code est bon .
346151	0.50	[L1]106,184,406;[L14]151-152;289-296[L2]47,262,319,436,489	Attention au style du code qui n'est pas cohérent tout au long du projet, les petites inattentions s'accroissent malheureusement.
346183	2.00	[L1]131, 148	bon travail ; warning indentation
346193	1.00	[L14]99 [L2]24, 25, 26, 27, 28, 99, 106, 112, 117, 121, 127, 140, 144, 151, 173	Attention à la limite de 87 caractères et à l'indentation. Dans l'ensemble, les conventions sont respectées.
346210	1.50	[L1]86,165,235,...;	Plutôt bon code dans l'ensemble. De manière générale évitez les boucles if sans aucune instruction derrière, il s'agit d'une mauvaise pratique (167,179,184,...). Essayez aussi de ne pas fermer les accolades sur la même ligne que le code, mise à part ça le style est bon!
346220	1.50	[L1]47, 55, 156	Très bonne indentation de manière générale, essayez de rester cohérent sur l'indentation quand vous avez une unique expression après un if ou un for
346228	2.00	[L2]515	Les conventions de programmation sont très bien respectées. Bravo !
346235	1.50	[L22]34,279	A quelques erreurs de distraction (probablement), les conventions de programmation sont respectées.
346247	2.00		Le style du code est homogène, aéré et très agréable à lire.
346255	1.50	[L2]56,219,220,228,247	Excellente indentation, style consistant. Dommage pour les lignes trop longues.
346301	1.00	[L1]39,103,300 ...; [L14]93,184 ;	Bon travail globalement. Attention toutefois à garder une indentation constante tout au long du code, et si possible indenter de la même manière fonctions et boucles conditionnelles. N'hésitez pas non plus à aérer un peu plus le code en ajoutant des espaces avant les accolades et des lignes vides de temps en temps. L'ajout de commentaires peut également aider à la compréhension lors de la relecture.
346370	1.00	[L1]60,94,127,140,... ; [L11]79	Le style est bien, le code est très (trop ?) aéré et bien lisible. Pour l'indentation, choisissez un style qui vous convient et respectez-le tout au long du programme. Trop de toutes petites fonctions finissent par nuire à la lisibilité.
346389	0.50	[L1] 134, 136, 138,... ; [L14] 12-15, 114-118; [L2] 137, 139	Le code n'est pas toujours correctement indenté et il y a des wrapping lines. Par contre, il est bien organisé et on s'y retrouve bien.

346450	0.00	[L1]50, 129, 155, 220, 224 [L14]91, 101:118, 244:254 [L2]31, 32, 143, 156, 158, 170, 173, ... [L22]187, 188	Beaucoup d'erreurs sont présentes, il faudrait revoir les conventions de programmation.
346471	1.00	[L14]159, 282, 290, 269, ...; [L22]249-250, 254-255	Style très soigné. Pas grand chose à redire. Dommage qu'il y ait quelques problèmes d'indentation et quelques problèmes d'alignement dans les conditions if.
346514	0.50	[L1] 317, 297 ; [L2]27,96,151,253 ; [L14]317,318 ;	Beau code dans l'ensemble ! Quelques erreurs vous coûtent malheureusement pas mal de point. Faites attention au respect des conventions sur la longueur des lignes, et faites particulièrement attention à l'indentation où vous faites quelques petites erreurs. Pensez aussi à laisser un peu d'espace entre vos fonction pour faciliter la lecture. N'hésitez pas à sortir les messages const string de vos fonction et de les mettre en global.
346515	2.00		Code très propre et très lisible. Bravo. Une ligne dépasse néanmoins la longueur de 87 caractères par ligne.
346518	2.00		Le style du code est très bon !
346544	2.00		Le code est très soigné, avec quelques commentaires mais pas trop pour très facilement naviguer dedans. Bravo !
346561	1.50	[L1]250, 253, 275, 280	Le syle est très soigné et le code très lisible. Attention à garder le même style d'accolades tout au long du programme. Attention également de bien respecter les conventions d'alignement lorsque qu'une ligne est reportée sur plusieurs lignes (non pénalisé).
346564	0.50	[L1]129,181,186,...;[L14]16- 372;[L2]10,48,51- 53,82,199,317	Attention à l'indentation qui varie (parfois 3, 4 ou 5 espaces) ou qui n'est pas présente du tout. Il vaut mieux utiliser des tabulations que des espaces pour que ce soit plus simple.
346565	2.00		Très bon style, bien lisible.
346576	1.50	[L22]48	Très bon style, code aéré et bien lisible. Peut-être un peu trop de ligne vides qui allongent le code pour rien.
346577			
346596	0.50	[L1]15,22,70,86,146; [L11]65,107; [L14]183,190	Les conventions sont à revoir. L'indentation et le style d'accolades n'est pas cohérent sur l'ensemble du code. Attention aux conventions sur les noms, autant pour les struct que pour les fonctions et les noms composés (code E). On place aussi généralement "using namespace std;" après tous les "#include". les noms des fonctions comme des variables sont préférablement parlants. Une relecture aurait été utile.
346718	1.00	[L14]153,154; [L2]74,434	Le code est en général bien structuré et clair. Le nom de certaines variables est peu clair (p.ex. le paramètre de la fonction calc_nbt qui s'appelle "mot" mais est un Dictionnaire).
346944	2.00		Indentation impeccable, le code est parfaitement lisible et bien commenté. Bravo !
346973	1.00	[L1]159-160, 77-78 ; [L14]176- 178, 242-243, 122-132; [L2] 206	Le code n'est pas toujours correctement indenté. Par contre, il est bien organisé et on s'y retrouve bien.
347035	1.50	[L22]122,125	Bonne indentation excepté right_place, vous pouvez revenir à la ligne pour une instruction unique après un if ou un for pour rendre la ligne plus lisible
347036	1.50	[L1]61, 103, 214	Très bonne indentation malgré un manque de consistance lors d'une instruction unique suivant un if

347038	2.00		Les conventions de programmation sont maitrisées.
347056	2.00		Attention aux indentations ligne 442. Au delà de ça, code impeccable, très belle maîtrise des conventions de style et les commentaires rendent le code extrêmement lisible et agréable à lire.
347057	2.00		
347087	1.00	[L1]106,137,183,...[L2]189,191,308,318	Il y a de nombreux problèmes d'indentations. Sinon, le reste va.
347105	2.00		L'effort consacré au code est remarquable, cependant il y a dans ce cas trop de commentaires et de lignes vides (le nombre de ligne peut largement être divisé par 2 juste en enlevant commentaires et lignes vides) L'organisation du fichier en générale n'est pas tout à fait correcte, il faut d'abord déclarer les typedefs, puis les variables globales avec constexpr, puis déclarer les fonctions, implémenter la fonction main et enfin implémenter les autres fonctions
347115	2.00		Excellente maîtrise des conventions de style. Petite remarque : plutôt que de mettre une if(condition) return true ou return false, on pourra simplement écrire "return condition"
347213	0.50	[L1] 74-76, 107, 141; [L14] 17-21, 88, 171; [L2] 24, 174, 209	Le code n'est pas toujours correctement indenté et il y a des wrapping lines. Autrement le style du code est plutôt facile à lire et bien espacé.
347262	2.00		Excellente indentation, le code est bien aéré et facile à lire
347263	0.00	[L1]84,157,234,235 ; [L11]56,132 ; [L14]318 ; [R2] 10,19	Le style global est loin d'être mauvais, cependant il est émaillé de petites erreurs qui vous coûtent des points, surtout au niveau de l'indentation, soyez prudent. Faites aussi très attention aux variables globales qui sont à proscrire dans tous les cas (sauf constantes) ! Evitez d'imbriquer trop de structure de contrôle les unes dans les autres, cela nuit à la lisibilité.
347288	1.00	[R2]14, [L14]567	Très bonne indentation malgré une légère faute ligne 567. L'utilisation des variables globales n'est néanmoins pas une bonne pratique.
347301			
347317			
347321	1.00	[L1]263-265, 288-290 [R2]20,38-41	Bon style dans l'ensemble, fais attention à ton style pour tes déclarations de contrôles qui n'est pas partout pareil. Il ne faut pas faire de variables globales.
347328	2.00		Attention à la limite de caractères qui a été dépassée une fois (ligne 385). Sinon les conventions de styles ont été respectées.
347341	1.50	[L2] 173, 224, 259	Le style du code est homogène, aéré et très agréable à lire. Par contre attention aux wrapping lines pour les commentaires.
347346	2.00		Les conventions sont très bien respectées ! Attention à la longue condition ligne 351, dans ce cas il faut couper en deux la ligne.
347369	1.50	[L1] 242-243, 277-279, 282	Le code est plutôt homogène et bien indenté à part quelques fautes. Pour améliorer le code la comparaison avec constantes char 'A' ou 'Z' plutôt que des valeurs brutes de code ASCII est préférable.
347381	2.00		Indentation impeccable, le code est parfaitement lisible et bien commenté. Bravo !
347394	1.00	[L1]99,100,111,185-190 ; [L11]251-272,290-291	Quelques erreurs d'indentation, dommage car le reste est très bien.

347428	1.50	[L1]63,197	Le code est en général clair et très bien structuré. Attention, les noms de fonctions doivent être en minuscule (ANAGRAM). Les commentaires sont utiles et clairs.
347445	2.00		Très bon style, indentation nickel et code agréable à lire. Nom de variable parlant dans l'ensemble.
347471			
347499	2.00		Parfait, de loin le meilleur projet qu'il m'ait été donné de corriger, avec mention spéciale pour la performance du code. Le seul conseil que j'ai à donner est d'essayer de créer une constante string "anagramme" qui est utilisée plusieurs fois en valeur brute dans le code, mais sinon c'est vraiment de l'excellent travail!
347510	1.00	[L1]84,85,94,98,147,198;[L14]91-95	Bon style, attention aux inattentions concernant les indentations!
347529			
347530	1.00	[L1] 55-56, 60-61, 318 ; [L14] 21-25, 116-132, 137-168	Le code n'est pas toujours correctement indenté. Il y a également beaucoup de valeurs hard-codées qu'en tant que lecteur on peine à comprendre -> pour améliorer faire comparaison avec constantes char 'A' ou 'Z' plutôt que des valeurs brutes de code ASCII.