

## Règles à respecter pour les TPs de SDD

### Remise du TP :

Le TP doit contenir :

- Le compte rendu du TP au format PDF,
- tout le code source, sans oublier le makefile. Ajoutez un README.txt si vous avez des commentaires particuliers à faire sur votre code.
- Tous les fichiers de tests que vous avez utilisés si le programme prend en entrée des fichiers.

Vous enverrez via Moodle votre travail, sans les fichiers temporaires (exécutable, \*.~, etc). Ne compressez pas votre TP. Un retard de rendu est forcément pénalisé. Après plusieurs jours de retard, il n'est plus possible d'envoyer votre travail.

### Points particuliers :

- Créez un fichier nommé 'main.c' pour le programme principal. C'est un nom générique et cela permet à l'enseignant qui vous corrige de savoir tout de suite où se trouve votre main. Ne mettez pas d'autres fonctions dans ce fichier, sauf recommandation contraire de l'enseignant.
- Créez toujours pour chaque structure une fonction afficher qui permet de parcourir une instance de la structure. Cette fonction permet de debugger plus rapidement votre code.

### Notation :

La note finale est sur 40 points, avec une amplitude de plus ou moins 10 points selon les Tps. L'exemple de notation suivante est là pour que vous compreniez comment vous allez être noté. Des différences peuvent exister selon les enseignants et selon les TPs donc ne négligez pas de les écouter si en début de cours ils donnent des consignes spécifiques.

Vous remarquerez qu'il y a autant de points attribués pour la partie codage des fonctions que pour la partie jeu d'essai.

Intitulé	Nb points	Commentaires
Présentation	1	La présentation comprend la description du TP (en 5 lignes minimum, comptez -0.25 par ligne manquante, sauf si votre texte est suffisamment complet en moins de 5 lignes), la qualité de la rédaction, etc.
Description SDD	1	Description par un schéma de toutes structures de données utilisées, fichiers d'entrée et de sortie inclus. Notez les noms de variable des structures.
Organisation des sources	1	Est noté ici la pertinence du découpage du programme en plusieurs fichiers et la cohérence des fonctions créées. La notation comprend également une partie rédaction : les fichiers et fonctions doivent être listés au début du rapport.
Fonctions	1 à 5	Le programme est noté par fonction. Au total, il y a entre 10 et 25 points

	points par fonctions	attribués. Pour chaque fonction, on attend que vous appliquiez les principes enseignés en SDD. Par exemple, la fonction 'creation' permettant d'allouer une structure doit vérifier que l'allocation s'est bien passée.
Entête fonctions et commentaires	2	Principe et commentaires pour chaque fonction et chaque fichier.
Lexique	2	Lexique des paramètres en entrée de fonction et des variables locales, rappel de l'ensemble des variables en en-tête du fichier.
Makefile	1	Le makefile doit fonctionner correctement et comprendre toutes les options de compilation que vous avez vues en cours de C. Pour les derniers Tps, le makefile doit également être générique.
Respect consignes	1	Le respect des consignes consiste à suivre le formatage demandé pour les entrées et les sorties, utiliser les principes vus en SDD, etc. Notez que si vous ne respectez rien, vous pouvez avoir ici des points négatifs.
Jeu d'essai	1 point par cas	Selon les Tps, il peut y avoir 10 à 25 cas. Présentez la liste des cas en sommaire de cette partie. Pour chaque cas il faut donner en titre le cas testé, puis l'entrée et la sortie du cas. Pour la sortie, vous pouvez par exemple montrer une capture d'écran de ddd ou du terminal.

A cela s'ajoute un décompte négatif de la note amenée sur 20 dans les cas suivants :

Intitulé	Nb points	Commentaires
Retard	0 pour tout le travail	Un travail rendu en retard est forcément pénalisé dès le premier jour de retard. Cette pénalisation suit une loi exponentielle qui dépend d'un TP à l'autre. Notez qu'au delà d'une semaine de retard, vous avez forcément 0 au TP.
Erreur de compilation	0 pour tout le travail	Aucune erreur de compilation n'est acceptée. Un travail rendu avec ce type d'erreur sera automatiquement noté 0.
Warning à la compilation	-1 par warning	Les warnings cachent souvent des erreurs graves de programmation. Vous ne devez donc pas en laisser dans votre code.
Erreur de segmentation et erreur valgrind	-1 à -5 par faute	En cas d'erreur de segmentation ou d'erreur valgrind, plusieurs pénalisations sont possibles selon les cas : <ul style="list-style-type: none"> <li>• si le cas existe dans le jeu d'essai <ul style="list-style-type: none"> <li>• si vous avez détecté l'erreur mais que vous n'avez pas réussi à la corriger, notez-le dans le cas du jeu d'essai avec également d'où vous pensez que ça provient. Vous aurez une pénalisation de -1.</li> <li>• si le rapport ne montre pas qu'il y a d'erreur, on en déduit que soit vous n'avez pas fait les tests valgrind, soit vous avez caché l'erreur. Vous aurez une pénalisation de -5.</li> </ul> </li> <li>• si le cas n'existe pas dans le jeu d'essai, vous aurez une pénalisation de -2.</li> </ul>
Copies	0 pour	Dans le cas où plusieurs binômes ont le même code et le même rapport, les

	chaque partie copiée	parties copiées sont notées 0. Vous pouvez vous aider entre binômes pour l'algorithmie mais cela ne peut pas vous donner exactement le même code.
Absence injustifiée	-7 à -10 points par absence	Les absences injustifiées ne sont pas tolérées car elle pénalise la personne avec laquelle vous travaillez et elle empêche l'enseignant de contrôler votre niveau.
Monôme		Les monômes ne sont pas autorisés. Néanmoins il peut en exister dans le cas d'un nombre impair dans le groupe ou si une personne d'un binôme ne vient plus en TP. Dans ce cas, la notation sera plus souple.