



Piscine C

Rush 01

*Résumé: Ce document est le sujet du Rush 01 de la piscine C de 42.*

*Version: 7.2*

# Table des matières

<b>I</b>	<b>Consignes</b>	<b>2</b>
<b>II</b>	<b>Préambule</b>	<b>3</b>
<b>III</b>	<b>Sujet</b>	<b>4</b>
<b>IV</b>	<b>Annexe</b>	<b>6</b>
<b>V</b>	<b>Rendu et peer-evaluation</b>	<b>8</b>

# Chapitre I

## Consignes

- Le groupe sera inscrit automatiquement en soutenance.
- Ne l'annulez pas, vous n'en aurez pas une deuxième.
- Toute demande de précision sur le sujet compliquera le sujet.
- Vous devez suivre la procédure de rendu pour le sujet.
- Le sujet peut changer jusqu'à une heure avant le rendu.
- Votre programme doit compiler avec les flags -Wall -Wextra -Werror, en utilisant le compilateur `cc`.
- Si votre programme ne compile pas, vous aurez 0.
- Votre programme doit être codé à la Norme. Si vous avez des fichiers ou fonctions bonus, celles-ci seront incluses dans la vérification de la norme et vous aurez 0 au projet en cas de faute de norme.
- Vous devrez donc réaliser le sujet indiqué avec les binômes imposés et vous présenter en soutenance à l'heure dite avec tous vos binômes.
- Lors de la soutenance, le projet devra être terminé. Les soutenances servent à présenter et à expliquer votre travail dans les moindres détails.
- Chaque membre du groupe devra parfaitement être au courant du travail réalisé, chacun des membres sera interrogé, la note du groupe étant basée sur les moins bonnes explications.
- Évidemment, vous devrez tout faire pour prendre contact avec vos binômes : téléphone, mail, pigeon voyageur, séance de spiritisme, etc. Aucune excuse ne sera acceptée en ce qui concerne les problèmes de groupe.
- Si après avoir vraiment tout essayé un de vos binômes reste injoignable : réalisez rush on s'arrangera en soutenance. Même si c'est le chef de groupe : vous avez tous accès au dépôt.
- Bon travail à tous.

# Chapitre II

## Préambule

Voici ce que Wikipédia a à dire à propos du paresseux à deux doigts :

Le paresseux à deux doigts est réputé être l'animal le plus lent au monde. L'adulte pèse en moyenne 4 à 8 kg et a la taille d'un petit chien : 60 à 85 cm de long avec une queue de 1,4 à 3,3 cm. Il présente un cou court, 4 longs membres d'égale longueur terminés par deux griffes incurvées à l'avant et trois à l'arrière ayant jusqu'à 7 cm de long. La tête est courte et plate, avec un nez retroussé, des oreilles rudimentaires et de grands yeux.

Ils passent près de 80 % de leur temps à dormir. Les paresseux se déplacent très lentement. Ils dorment, mangent, s'accouplent, donnent naissance et élèvent leurs petits sans jamais descendre de leur arbre, dans lequel ils restent accrochés aux branches, la tête en bas. Ils ne vont que très rarement à terre, pour changer d'arbre quand la nourriture s'y fait rare ou pour déféquer, une fois par semaine.


Ils sont parfaitement camouflés dans les arbres, leur corps brun verdâtre ramassé sur lui-même passant facilement pour un nid de termites ou une excroissance de bois au yeux des prédateurs. En cas d'attaque, ils se défendent avec dents et griffes bien qu'ils soient d'un tempérament pacifique, comptant plus sur leur capacité à se fondre dans le décor. Ils restent immobiles pendant de longues heures mais se grattent régulièrement, trahissant ainsi parfois leur présence.

Contrairement aux paresseux, vous n'allez pas avoir beaucoup de temps pour dormir ce week-end.

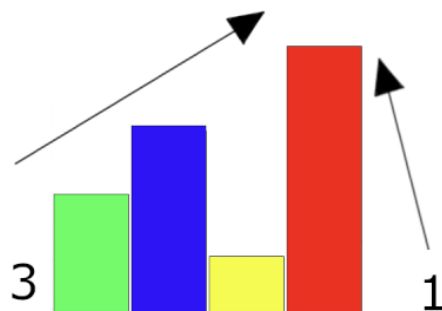
Quel dommage.

# Chapitre III

## Sujet

	Exercice : 00
rush-01	
Dossier de rendu : <i>ex00/</i>	
Fichiers à rendre : <b>Tous les fichiers nécessaires</b>	
Fonctions Autorisées : <b>write, malloc, free</b>	

- Votre code source sera compilé par la commande : `cc -Wall -Wextra -Werror -o rush-01 *.c`
- Votre dossier de rendu devra contenir tout ce qui sera nécessaire à la compilation de votre programme.
- Créez un programme qui résoud le problème suivant :
- Sur une map de 4 par 4, placez des caisses de taille 1 à 4 de manière à ce que chaque ligne et colonne voie le bon nombre de caisses de chaque point de vue possible.
- Exemple : la caisse de taille 3 cache ici la caisse de taille 1, ce qui fait qu'il n'y a que 3 boîtes visibles de la gauche. De la droite, la caisse de taille 4 cache toutes les autres caisses, il y a donc une seule caisse de visible.



- Chacun des vues (deux par ligne et deux par colonne) aura une valeur donnée,

allant de 1 à 4. Votre programme doit placer les caisses correctement, tout en faisant attention à n'avoir qu'une seule caisse de chaque hauteur sur chaque ligne et colonne.

- Si il y a plusieurs solution, vous devez afficher la première que vous trouverez.
- Le programme sera lancé de la façon suivante :

```
> ./rush-01 "col1up col2up col3up col4up col1down col2down col3down col4down row1left row2left  
row3left row4left row1right row2right row3right row4right"
```

- (cf. annexe 1)
- col1up correspond à la valeur pour le point de vue du haut de la colonne de gauche. Chaque valeur doit être comprise entre 1 et 4.
- Vous ne devez accepter que les inputs respectant ce format. Tout autre format doit être considéré comme une erreur.
- Vous devrez afficher en sortie la résolution comme suit :

```
> ./rush-01 "4 3 2 1 1 2 2 2 4 3 2 1 1 2 2 2" | cat -e  
1 2 3 4$  
2 3 4 1$  
3 4 1 2$  
4 1 2 3$
```

- (cf. annexe 2 et 3)
- En cas d'erreur ou si vous ne trouvez pas de solution, vous écrirez seulement "Error" suivi d'un retour à la ligne.
- Si vous voulez des points bonus essayez de gerer d'autre taille de map (jusqu'à 9x9 ??).
- Si la partie obligatoire ne passe pas les tests vous aurez 0.

# Chapitre IV

## Annexe

Ce qui suit est une représentation artistique de votre programme. Vous devez, bien entendu, respecter les consignes de rendu telles que décrites dans la partie précédente. Ces visualisations ont pour unique but de vous aider à appréhender ce sujet.

- Annexe 1 :

	col1up	col2up	col3up	col4up	
row1left					row1right
row2left					row2right
row3left					row3right
row4left					row4right
	col1down	col2down	col3down	col4down	

- Représentation du lancement du programme avec col\_up, col\_down, row\_left et row\_right
- Annexe 2 :

	4	3	2	1	
4					1
3					2
2					2
1					2
	1	2	2	2	

- En remplaçant les col\* et row\*, on obtient ceci.
- Annexe 3 :

	4	3	2	1	
4	1	2	3	4	1
3	2	3	4	1	2
2	3	4	1	2	2
1	4	1	2	3	2
	1	2	2	2	

- Votre programme doit remplir les cases intérieures, et renvoyer la réponse tel que demandé dans le sujet.



# Chapitre V

## Rendu et peer-evaluation

Rendez votre travail sur votre dépôt `Git` comme d'habitude. Seul le travail présent sur votre dépôt sera évalué en soutenance. Vérifiez bien les noms de vos dossiers et de vos fichiers afin que ces derniers soient conformes aux demandes du sujet.

Vu que votre travail ne sera pas évalué par un programme, organisez vos fichiers comme bon vous semble du moment que vous rendez les fichiers obligatoires et respectez les consignes du sujet.



**Vous ne devez rendre uniquement les fichiers demandés par le sujet de ce projet.**