



## Compte rendu du travail mené par le groupe

Lors de notre licence 3 : Science du numérique, nous avons eu l'opportunité de travailler en groupe sur des problèmes sortis à l'ICPC les années précédentes.

Nous avons pu à travers cela découvrir des techniques de programmation avancée dans le langage Python mais aussi de renforcer des acquis l'algorithmie.

Nous avons pu, à travers les séances de Ac'Lab résoudre 13 problèmes dispatchés sur plusieurs années du concours.

Les thématiques abordées sont vastes, traitant des mathématiques à la biologie en passant par des problèmes humains.

Il nous est arrivé de rester bloqués sur certain problème, les difficultés rencontrées n'étaient pas forcément techniques, il s'agissait souvent d'une incompréhension du sujet initial mais il arrivait quelque fois qu'il s'agisse d'un souci de mise en œuvre de l'idée exposée.

## Table des matières

Compte rendu du travail mené par le groupe .....	1
Année 1993 .....	3
Problème A [Budget Travel] .....	3
Problème B .....	3
Problème F [Puzzle] .....	3
Année 1994 .....	3
Problème B [Testing the CATCHER] .....	3
Année 1995 .....	4
Problème E .....	4
Problème H [Uncompress] .....	4
Année 1996 .....	4
Problème B [Calling circles] .....	4
Problème E .....	4
Problème F [Pattern Matching Prelims] .....	5
Année 1997 .....	5
Problème B [Jill Rides Again] .....	5
Année 1999 .....	5
Problème H [Flooded!] .....	5
Année 2000 .....	5
Problème B .....	5
Année 2015 .....	5
Problème A .....	5

## Année 1993

### Problème A [Budget Travel]

Ce programme a été réalisé par Johann.

La résolution de ce problème était de trouver pour un trajet soumis à une série de paramètres, son moindre coût.

Il n'a pas été aisé de trouver la valeur exacte montrée dans l'énoncé, j'ai tout de même réussi à parvenir à une valeur approchée, je ne sais pas si ma méthode de calcul n'est pas optimale où si je remonte mal mes données mais dans tous mes essais pour y parvenir je suis tombé sur ces mêmes valeurs.

Il s'agissait du premier problème que j'ai pu faire, j'ai eu énormément de difficultés pour construire un parseur efficace car je n'avais de notion de python que les restes de DUT 1<sup>ère</sup> année où même là nous n'avions aborder que les prémices du langage.

Le nom du programme : ICPC\_1993v2.py

(ICPC\_1993.py était un premier jet)

A modifier dans le programme avant lancement :

- `os.chdir()` : où se trouve le fichier à remonter
- `parcourir_fichier("fichier.txt_old")` : le nom du fichier à remonter

### Problème B

Ce programme a été réalisé par Gaëtan.

Le but de ce problème est de rédiger un programme qui prendra en entrée les données de l'arpenteur et produira en sortie les informations souhaitées sur la nature des lots dans le lotissement.

### Problème F [Puzzle]

Ce programme a été réalisé par Tristan. Voilà.

## Année 1994

### Problème B [Testing the CATCHER]

Ce programme a été réalisé par Sébastien.

J'ai mis un peu de temps à comprendre et c'est aussi celui que j'ai fait en dernier. Je n'ai, hélas, pas eu le temps de le finir. Je l'ai quand même laissé afin que vous voyiez l'avancée de ce problème. Le but de ce sujet est d'offrir une vérification des performances du missile CATCHER.

Pour le moment, le programme ne contient qu'une boucle infinie qui accepte des valeurs. Pour finir un test il suffit de rentrer -1 et si la première valeur du test est -1 alors le programme s'arrête.

## Année 1995

### Problème E

Gaëtan

### Problème H [Uncompress]

Ce programme a été réalisé par Johann.

Il consistait en la modification d'un texte remonté d'un fichier.

Le texte comprenait divers entiers en son sein et ceux-ci devaient être remplacés par les n mots qui les précédaient où n valait l'entier trouvé.

J'ai eu énormément de plaisir à faire ce problème car ce fût le premier problème où j'ai trouvé le résultat exact attendu.

J'ai eu quelques difficultés pour replacer les ponctuations que j'enlève à la lecture du fichier dans mon parseur.

Le nom du programme : ICPC\_1995.py

A modifier dans le programme avant lancement :

- `os.chdir()` : où se trouve le fichier à remonter
- `parcourrir_fichier("H_1995.txt")`: le nom du fichier à remonter

## Année 1996

### Problème B [Calling circles]

Tristan

### Problème E

Gaëtan

## Problème F [Pattern Matching Prelims]

Tristan

Année 1997

## Problème B [Jill Rides Again]

Ce programme a été réalisé par Sébastien.

J'ai décidé de le réaliser en Python. Je trouvais que c'était une bonne opportunité d'exercer ma maîtrise de ce langage. J'ai trouvé ce sujet particulièrement intéressant. Le but étant de donner, pour chaque partie d'un trajet, le fragment de la route qui est la plus jolie.

Au tout début, l'utilisateur doit mettre le nombre de route qu'il compte rentrer dans le programme. Il doit ensuite mettre les notes qu'il a attribué aux différents fragments de route. Le programme indique ensuite les parties de la route les plus jolies.

Année 1999

## Problème H [Flooded!]

Ce programme a été réalisé par Sébastien.

J'ai réalisé ce problème en C. Cependant, j'ai mis pas mal de temps à comprendre le sujet. Cependant, une fois lancer dedans, je voulais absolument le réussir. Je me suis servi de diverses sources afin de voir comment je pouvais construire le code.

Année 2000

## Problème B

Sébastien

Année 2015

## Problème A [Amalgamated Artichokes]

Ce programme a été réalisé par Johann.

Ce programme consistait en déterminer la plus grande descente de prix à l'aide d'un input saisi en entrée et d'une formule mathématique :

$$\text{price}(k) = p \cdot (\sin(a \cdot k + b) + \cos(c \cdot k + d) + 2)$$

Où  $p$  est le prix de départ,  $a$ ,  $b$ ,  $c$  et  $d$  des constantes saisies en entrée et  $k$  le cycle (variant de 1 à  $n$  lui aussi saisi en entrée).

Je pense qu'il s'agit là du problème le plus simple que j'ai dû résoudre au cours de ces séances.

A contrario des autres, le sujet était assez simple à comprendre et l'algorithme facile à écrire, j'ai beaucoup aimé travailler sur ce sujet.

Le nom du programme : ICPC\_2015.py

A modifier dans le programme avant lancement :

- NA

A faire au lancement :

- ✓ Fournir lorsqu'il est demandé une ligne dans ce genre : 42 1 23 4 8 10 ( $p$   $a$   $b$   $c$   $d$   $n$ )

Où  $P$  doit être compris entre 1 et 1000 inclus.

Où  $A, B, C$  et  $D$  doivent être compris entre 0 et 1000 inclus.

Et où  $N$  doit être compris entre 1 et  $10^6$  inclus.