DRAPIED Hugo 3A

Document individuelle
SAE S1.01

1/Pour le programme **main** :
1.Glossaire

Identificateur	Type	Rôle
i	int	Compteur permettant de parcourir le tableau dans le sens de la longueur (Tab[i][])
j	int	Compteur permettant de parcourir le tableau dans le sens de la largeur (Tab[][j]
Tableau	int[][]	Tableau en 2 dimensions contenant les données des participants
TableauClassement	int[][]	Tableau en 2 dimensions contenant le temps et le numéro des trois meilleurs participants
NbPorte	int	Nombre de portes dans la course
reponse	String	Variable recevant la réponse à si l'utilisateur veut le temps en minutes et secondes
joueurs	int	Nombre de joueurs participants à la course (longueur du Tableau)
premier	int	Numéro de brassard du premier
deuxieme	int	Numéro de brassard du deuxième
troisieme	int	Numéro de brassard du troisième
TempsP	int	Temps du premier de la course
TempsD	int	Temps du deuxième de la course
TempsT	int	Temps du troisième de la course
Toucher	int	Nombre de portes touchées par un participant
Oubli	int	Nombre de portes oubliées par un participant
Somme	int	Somme des erreurs effectués (test somme ≤ NbPorte)
Erreur	int	Nombre de points d'erreur finaux selon les oublies et les touches

minute	int	Temps en minute de chaque participant (entier)
seconde	int	Reste du temps en seconde de chaque participant (à mettre avec minutes pour le temps total)
pfBorneInf	int	Valeur min bornant l'entrée de temps par manche
pfBorneSup	int	Valeur max bornant l'entrée de temps par manche
Saisie	String	Permet de vérifier si on a rentré « o » ou « oui » lors de la confirmation de manche
Minutes1	int	Nombre de minutes du temps fait par le premier
Minutes2	int	Nombre de minutes du temps fait par le deuxième
Minutes3	int	Nombre de minutes du temps fait par le troisième
Secondes1	int	Nombre de secondes du temps fait par le premier (sans les minutes)
Secondes2	int	Nombre de secondes du temps fait par le deuxième (sans les minutes)
Secondes3	int	Nombre de secondes du temps fait par le troisième (sans les minutes)

2. Commentaire Java détaillé

```java
/**
 * SAE Implementation.
 *
 * DRAPIED Hugo
 */
import java.util.Scanner;
public class CourseCanoë {
    public static void main(String args[]){
        Scanner clavier = new Scanner(System.in); //initialisation Scanner
        String reponse=""; //initialisation variable reponse
        int NbPorte=Nbporte(); //initialisation de NbPorte puis appel Nbporte()
        int[][] Tableau; //initialisation du tableau en 2 dimensions
        Tableau=CreerTableau(); // le tableau prend les dimensions souhaités via appel CreerTableau
        saisieTableau(Tableau, NbPorte);//Remplissage des lignes 0-6 tableau via saisieTableau
        TempsFinal(Tableau);//Remplissage de la ligne numéro 7 du tableau
        AfficherTemps(Tableau);//Affiche le temps en millisecondes pour chaque joueur
        int[][] TableauClassement;//initialisation du tableau servant pour le podium
        TableauClassement= CreerTabClass(); //on donne les dimensions souhaités au tableau
        Classement(Tableau,TableauClassement);//On remplit le tableau du classement
        AfficherPodium(TableauClassement);//On affiche le résultat du podium présent dans le tableau
        System.out.println("Afficher résultats et podium en minutes secondes ? (oui/non)");
        reponse=clavier.nextLine();
        switch(reponse){ //On vérifie si la personne a rentré "oui",
                case "oui" -> { //si c'est le cas on affiche le temps en minutes et secondes
                    ConversionTemps(Tableau); //affichage des résultats globaux en min sec
                    PodiumMinSec(TableauClassement);//affichage du podium en min sec
                }
        }
    }
```

3.Explication

-On initialise les variables tableau et tableauClassement et NbPorte

-On obtient la valeur de NbPorte, on met aux bonnes dimensions puis on remplit entièrement les deux tableaux (d'abord Tableau, puis TableauClassement)

-Affichage des temps effectués par tous les participants et du podium, les trois meilleurs ainsi que leurs temps

-On demande si l'utilisateur veut obtenir les mêmes données que plus haut mais en minutes et secondes au lieu de millisecondes, s'il répond oui on les affiche, sinon le programme s'arrête

DEBUT

reponse ← « »
NbPorte ← Nbporte()
Tableau ← CreerTableau()
TableauClassement ← CreerTabClass()
SaisieTableau(Tableau,NbPorte)
TempsFinal(Tableau)
AfficherTemps(Tableau)
Classement(Tableau,TableauClassement)
AfficherPodium(TableauClassement)
AFFICHER « Afficher résultats et podiums en minutes secondes ? (oui/non) »
Saisir reponse
Si reponse = « oui » ALORS
    ConversionTemps(Tableau)
    PodiumMinSec(TableauClassement)
FIN SI

FIN

4. Jeux d’essai

Variable	Valeur	Résultat attendu
NbPorte	18	Le participant n°1 a fait un temps de 240000 millisecondes. Le participant n°2 a fait un temps de 220000 millisecondes. Le participant n°3 a fait un temps de 211000 millisecondes. La première place revient au numéro 3 avec un temps de 211000 millisecondes. La deuxième place revient au numéro 2 avec un temps de 220000 millisecondes. La troisième place revient au numéro 1 avec un temps de 240000 millisecondes. Le participant n°1 a fait un temps de 4m et 0s. Le participant n°2 a fait un temps de 3m et 40s. Le participant n°3 a fait un temps de 3m et 31s. La première place revient au numéro 3 avec un temps de 3min et 31sec . La deuxième place revient au numéro 2 avec un temps de 3min et 40sec . La troisième place revient au numéro 1 avec un temps de 4min et 0sec .
joueur	3	
Touche	Joueur 1 : manche1= 4 /manche2=1 Joueur 2 : manche1=3 /manche2=2 Joueur 3 : manche1= 1 /manche2=5	
Oubli	Joueur 1 : manche1= 0 /manche2=1 Joueur 2 : manche1= 0 /manche2=0 Joueur 3 : manche1= 0 /manche2=0	
Tableau	{1, 1, 100000, 8, 1, 80000, 52, 0}, {2, 1, 120000, 6, 1, 90000, 4, 0}, {3, 1, 94000, 2, 1, 105000, 10,0}	
TempsFinal	{1, 1, 100000, 5, 1, 80000, 3, 188000}, {2, 1, 120000, 6, 1, 90000, 4, 220000}, {3, 1, 94000, 2, 1, 105000, 10,211000}	
pfBorneSup	240 000	
pfBorneInf	0	
Reponse	« oui »	

```
Combien de portes comporte votre course ?
18
Rentrez le nombre de participants, il doit y en avoir entre 2 et 49.
3
Le joueur numéro 1 a-t-il validé la manche 1 ? (si oui : saisissez oui ou o)
o
Veuillez saisir le temps (en ms) parcouru à la première manche :
Donnez une valeur comprise entre 0 et 240000.
100000
Combien de portes le joueur a-t-il touché ?
4
Combien de portes le joueur a-t-il oublié ?
0
Le joueur numéro 2 a-t-il validé la manche 1 ? (si oui : saisissez oui ou o)
o
Veuillez saisir le temps (en ms) parcouru à la première manche :
Donnez une valeur comprise entre 0 et 240000.
120000
Combien de portes le joueur a-t-il touché ?
3
Combien de portes le joueur a-t-il oublié ?
0
Le joueur numéro 3 a-t-il validé la manche 1 ? (si oui : saisissez oui ou o)
o
Veuillez saisir le temps (en ms) parcouru à la première manche :
Donnez une valeur comprise entre 0 et 240000.
94000
Combien de portes le joueur a-t-il touché ?
1
Combien de portes le joueur a-t-il oublié ?
0
Le joueur numéro 1 a-t-il validé la manche 2 ? (si oui : saisissez oui ou o)
o
Veuillez saisir le temps (en ms) parcouru à la deuxième manche :
Donnez une valeur comprise entre 0 et 240000.
80000
Combien de portes le joueur a-t-il touché ?
1
Combien de portes le joueur a-t-il oublié ?
1
```

```
Combien de portes le joueur a-t-il oublié ?
1
Le joueur numéro 2 a-t-il validé la manche 2 ? (si oui : saisissez oui ou o)
o
Veuillez saisir le temps (en ms) parcouru à la deuxième manche :
Donnez une valeur comprise entre 0 et 240000.
90000
Combien de portes le joueur a-t-il touché ?
2
Combien de portes le joueur a-t-il oublié ?
0
Le joueur numéro 3 a-t-il validé la manche 2 ? (si oui : saisissez oui ou o)
o
Veuillez saisir le temps (en ms) parcouru à la deuxième manche :
Donnez une valeur comprise entre 0 et 240000.
105000
Combien de portes le joueur a-t-il touché ?
5
Combien de portes le joueur a-t-il oublié ?
0
Le participant n°1 a fait un temps de 240000 millisecondes.
Le participant n°2 a fait un temps de 220000 millisecondes.
Le participant n°3 a fait un temps de 211000 millisecondes.

La première place revient au numéro 3 avec un temps de 211000 millisecondes.

La deuxième place revient au numéro 2 avec un temps de 220000 millisecondes.

La troisième place revient au numéro 1 avec un temps de 240000 millisecondes.

Afficher résultats et podium en minutes secondes ? (oui/non)
oui
Le participant n°1 a fait un temps de 4m et 0s.
Le participant n°2 a fait un temps de 3m et 40s.
Le participant n°3 a fait un temps de 3m et 31s.

La première place revient au numéro 3 avec un temps de 3min et 31sec .
La deuxième place revient au numéro 2 avec un temps de 3min et 40sec .
La troisième place revient au numéro 1 avec un temps de 4min et 0sec .
```

2/Pour le programme **NbPorte** :

1.Glossaire

Identifiant	Type	Rôle
NbPorte	int	Nombre de portes dans la course

2.Commentaire Java détaillé

```java
public static int Nbporte(){
    int Porte=0;
    Scanner clavier = new Scanner(System.in);
    System.out.println("Combien de portes comporte votre course ?");
    Porte = clavier.nextInt();
    while (Porte <18 || Porte >22){ // On contrôle que le nombre de portes dans le parcours est entre 18 et 22
        System.out.println("Incorrect, votre parcours doit comporter entre 18 et 22 portes.");
        Porte = clavier.nextInt();
    }
    return Porte;
}
```

3.Explication

-On initialise porte à 0

-On demande à saisir porte puis on test la condition de la boucle

-Si la condition est respecté on retourne directement porte sinon on rentre dans la boucle

-On affiche incorrect, et on redemande à ressaisir porte, tant que porte n'est pas compris entre 18 et 22 on reste dans la boucle

-On finit en retournant porte

DEBUT

SAISIR Porte
TANT QUE Porte >22 OU Porte < 18 FAIRE
AFFICHER « Incorrect, votre parcours doit comporter entre 18 et 22 portes. »
SAISIR Porte
FIN TANT QUE
RETOURNER Porte
FIN

4. Jeux d'essai

Porte	Résultat attendu
4	Incorrect, votre parcours doit comporter entre 18 et 22 portes. (→saisir de nouveau porte)
25	Incorrect, votre parcours doit comporter entre 18 et 22 portes. (→ saisir de nouveau porte)
21	(retourner porte) 21

1er cas :

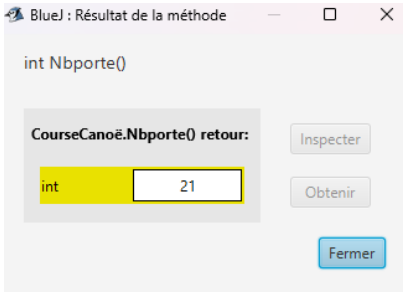
```
Combien de portes comporte votre course ?
4
Incorrect, votre parcours doit comporter entre 18 et 22 portes.
```

2ème cas :

```
Combien de portes comporte votre course ?
25
Incorrect, votre parcours doit comporter entre 18 et 22 portes.
```

3ème cas :

```
Combien de portes comporte votre course ?
21
```



Explication :

On contrôle que le nombre de portes dans le parcours soit entre 18 et 22

3/Pour le sous programme **CreerTableau** :

1.Glossaire

Identifiant	Type	Rôle
Tableau	Int[][]	Est la variable initiale du tableau qui sera modifiée puis renvoyée
Joueur	Int	Longueur du tableau (nombre de participants)

2.Commentaire Java détaillé

```
public static int[][] CreerTableau() {
    int tableau[][];
    Scanner clavier = new Scanner(System.in);
    int joueur=0;
    while (joueur >=50 || joueur<=1){ //On contrôle que le nombre de joueurs est bien compris entre 2 et 49
        System.out.println("Rentrez le nombre de participants, il doit y en avoir entre 2 et 49.");
        joueur = clavier.nextInt();
    }
    tableau = new int[joueur][8]; //On donne au tableau autant de colonnes que de joueurs et 8 lignes
    return tableau;

}
```

3.Explications

-On initialise le tableau vide et sans dimensions

-On demande de saisir le nombre de joueurs participant à la course

-On vérifie que le nombre de joueurs est compris entre 2 et 49, si ce n'est pas le cas on refait saisir jusqu'à ce que ça le soit

-On donne au tableau les dimensions suivantes : nombre de colonnes = joueur, nombre de lignes = 8

-On retourne le tableau

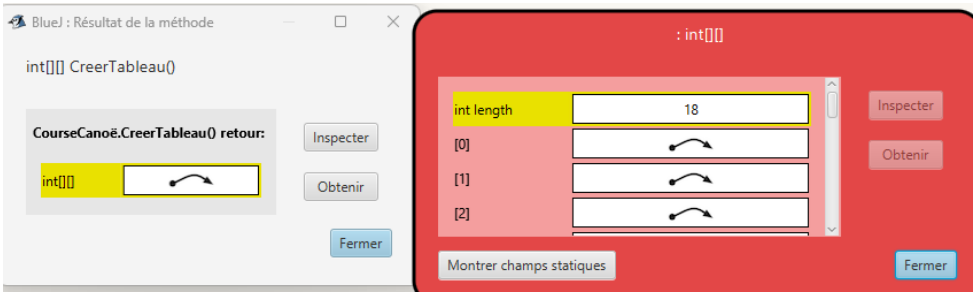
DEBUT
joueur ←0
TANT QUE joueur >=50 OU joueur <=1 FAIRE
AFFICHER « Rentrez le nombre de participants, il doit y en avoir entre 2 et 49 .»
SAISIR joueur
FIN TANT QUE
Tableau[][] ← Tableau[joueur][8]
RETOURNER Tableau
FIN

4.Jeux d’essai

Joueur	Résultat attendu
18	Tableau[18][8]
1	Rentrez le nombre de participants, il doit y en avoir entre 2 et 49 . (→ressaisir joueur)
50	Rentrez le nombre de participants, il doit y en avoir entre 2 et 49 . (→ressaisir joueur)

1er cas :

```
Rentrez le nombre de participants, il doit y en avoir entre 2 et 49.
18
```



2ème cas :

```
Rentrez le nombre de participants, il doit y en avoir entre 2 et 49.
1
Rentrez le nombre de participants, il doit y en avoir entre 2 et 49.
```

3ème cas :

```
Rentrez le nombre de participants, il doit y en avoir entre 2 et 49.
50
Rentrez le nombre de participants, il doit y en avoir entre 2 et 49.
```

Explication :

On contrôle que le nombre de joueurs dans le parcours soit entre 2 et 49

4/Pour le sous programme **NbErreur** :

1.Glossaire

Identifiant	Type	Rôle
Toucher	int	Nombre de portes touchées par le joueur
Oubli	int	Nombre de portes oubliées par le joueur
Somme	int	Nombre d'erreurs commis par le joueur
Erreur	int	Nombre de points d'erreur du joueur
PfPorte	Int	Nombre de porte dans la course

2.Commentaire Java détaillé

```java
public static int NbErreur(int pfPorte){
    Scanner clavier = new Scanner(System.in);
    int Toucher=0;
    int Oubli=0;
    int Somme=-1;
    while (Somme>pfPorte || Somme<0){ //On contrôle que le nombre d'erreurs entre 0 et pfPorte (nombre de portes du parcours)
        System.out.println("Combien de portes le joueur a-t-il touché ?");
        Toucher= clavier.nextInt();
        System.out.println("Combien de portes le joueur a-t-il oublié ?");
        Oubli=clavier.nextInt();
        Somme=Toucher+Oubli;
        if (Somme > pfPorte){ //On contrôle que le nombre d'erreur n'est pas supérieur au nombre de portes, si oui on envoie ce message
            System.out.println("Le joueur ne peut pas faire plus d'erreur qu'il n'y a de portes, recommencez : \n");
        }
    }
    int Erreur;
    Erreur=Oubli*50 + Toucher*2;
    return Erreur;
}
```

3.Explications

-On initialise Toucher, Oubli à 0 puis Somme à -1

-Dans la boucle on demande à saisir Toucher et Oubli

-si somme est supérieur à pfPorte et positif, on affiche "Le joueur ne peut pas faire plus d'erreur qu'il n'y a de portes, recommencez : »

-si somme est inférieur à pfPorte et est positif, on sort de la boucle

-Ensuite on calcul Erreur selon Oubli et Toucher

-On retourne Erreur

DEBUT
Somme← -1
TANT QUE Somme>pfPorte OU Somme<0 FAIRE
    AFFICHER « Combien de portes le joueur a-t-il touché ? »
    SAISIR Toucher
    AFFICHER «Combien de portes le joueur a-t-il oublié ? »
    SAISIR Oublier
    SI Somme>pfPorte ALORS
        AFFICHER «Le joueur ne peut pas faire plus d'erreur qu'il n'y a de portes, recommencez »
    FIN SI
FIN TANT QUE
*Erreur ← Oublier*50 + Toucher*2*
Retourner Erreur
FIN

4.Jeux d'essai

Variable	Valeur	Résultat attendu
Toucher	1er :14 | 2ème :5	Premier cas : "Le joueur ne peut pas faire plus d'erreur qu'il n'y a de portes, recommencez : » (→ressaisit de oubli et toucher) 2ème cas: RETOURNER Erreur (erreur=110)
Oubli	1er :5 | 2ème: 2	
Somme	1er :19 | 2ème :7	
PfPorte	1er :18 | 2ème : 20	

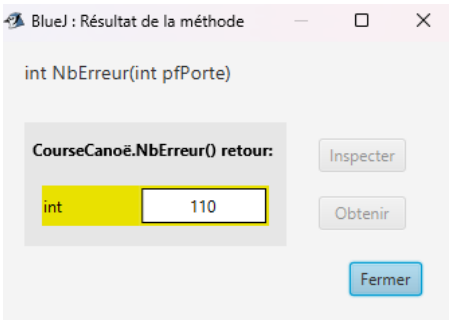
1er cas :

```
Combien de portes le joueur a-t-il touché ?
14
Combien de portes le joueur a-t-il oublié ?
5
Le joueur ne peut pas faire plus d'erreur qu'il n'y a de portes, recommencez :

Combien de portes le joueur a-t-il touché ?
```

2ème cas :

```
Combien de portes le joueur a-t-il touché ?
5
Combien de portes le joueur a-t-il oublié ?
2
```



Explications :

On contrôle que le nombre d'erreurs soit inférieurs au nombre de porte, puis on renvoie le nombre de points d'erreurs qui correspond au nombre et au type d'erreurs commises.

5/ Pour le sous programme **AfficherTemps** :

1.Glossaire

Identifiant	Type	Rôle
pfTableau	int[][]	Tableau contenant (entre autre) le numéro des joueurs et le temps compensé final
Joueur	Int	Nombre de participants
I	Int	Compteur

2.Commentaire Java détaillé

```
public static void AfficherTemps(int[][] PfTableau){
    /*Scanner clavier = new Scanner(System.in);
    int[][] PfTableau = new int [2][8];
    for (int i=0; i<2;i++){
        for (int j=0; j<8; j++){
            PfTableau[i][j]=clavier.nextInt();
        }
    }*/
    int joueur = PfTableau.length;
    /* On parcourt le tableau, pour chaque colonne on vérifie si le temps composé est égal à 0 ou pas, si il l'est on renvoie que le joueur tab[i][j=0]
    n'a pas fini la course. Sinon on renvoit que le joueur tab[i][j=0] a fait pour temps tab[i][j=7] */
    for (int i=0; i<joueur; i++){
        if (PfTableau[i][7]==0){
            System.out.println("Le participant n°"+PfTableau[i][0]+" n'a pas fini la course.");
        } else {
            System.out.println("Le participant n°"+PfTableau[i][0]+" a fait un temps de "+PfTableau[i][7]+" millisecondes.");
        }
    }
    System.out.println("");
```

3.Explications

-On initialise joueur avec PfTableau .length

-On rentre dans une boucle nous faisant parcourir le tableau colonne par colonne

-À chaque nouvelle colonne on teste si PfTableau[i][7] est égal à 0, si c'est le cas c'est que ce participant n'a pas fini la manche 1 ou la manche 2, on affiche alors « Le participant n°(numéro du participant) n'a pas fini la course. »

-Si PfTableau[i][7] n'est pas égal à 0, c'est que le participant à fini la course, on affiche alors « Le participant n°(numéro du participant) a fait un temps de (temps du participant) millisecondes."

DEBUT

```
joueur ← pfTableau.length
POUR (INIT i←0 ; TEST: i<joueur ; MAJ : i++ ) FAIRE
        SI PfTableau[i][7] = 0 ALORS
                AFFICHER «Le participant n°"+PfTableau[i][0]+" n'a pas fini la course.»
        SINON
                AFFICHER "Le participant n° (PfTableau[i][0])  a fait un temps de (pfTableau[i][7] millisecondes."
        FIN SI
FIN POUR
FIN
```

4.Jeux d’essai

pfTableau	Résultat attendu
{1, 1, 100000, 8, 1, 80000, 52, 240000}, {2, 1, 120000, 6, 1, 90000, 4, 220000}, {3, 1, 94000, 2, 0, -, -, 0}	Le participant n°1 a fait un temps de 240000 millisecondes. Le participant n°1 a fait un temps de 220000 millisecondes. Le participant n°3 n’a pas fini la course.

```
Le participant n°1 a fait un temps de 240000 millisecondes.
Le participant n°2 a fait un temps de 220000 millisecondes.
Le participant n°3 n'a pas fini la course.
```

Explication :

On récupère le temps compensé total de chaque joueur systématiquement présent de la ligne j=7 (8ème ligne). On vérifie si elle est non nulle, si c’est le cas on affiche le numéro du joueur et son temps, sinon cela veut dire qu’il n’a pas fini la course.

6/Pour le sous programme **ConversionTemps** :

1.Glossaire

Identifiant	Type	Rôle
minute	int	Temps en minute de chaque participant (entier)
seconde	int	Reste du temps en seconde de chaque participant (à mettre avec minutes pour le temps total)
joueur	int	Nombre de participants
PfTableau	int[][]	Tableau contenant (entre autre) le numéro des joueurs et le temps compensé final

2. Commentaire Java détaillé

```java
public static void ConversionTemps(int[][] PfTableau) {
    /*Scanner clavier = new Scanner(System.in);
    int[][] PfTableau = new int [2][8];
    for (int i=0; i<2;i++){
        for (int j=0; j<8; j++){
            PfTableau[i][j]=clavier.nextInt();
        }
    }*/
    int joueur = PfTableau.length;
    int minute=0;
    int seconde=0;
    for (int i=0; i<joueur; i++){
        if (PfTableau[i][7]!=0){
            minute = (PfTableau[i][7]/1000)/60;
            seconde = (PfTableau[i][7]/1000)%60;
            System.out.println("Le participant n°"+PfTableau[i][0]+" a fait un temps de "+minute+"m et "+seconde+"s.");
        }
    }
    System.out.println("");
}
```

3. Explications

-On initialise joueur = pfTable.length ainsi que minute et seconde à 0

-On parcourt toute la table colonne à colonne

-Si pfTableau[i][7] n'est pas égale à 0, alors on calcul les minutes et secondes

-On affiche ensuite "Le participant n°(numéro du participant) a fait un temps de (minute)m et (seconde)s."

```
DEBUT
joueur ← pfTableau.length
minute←0
seconde←0
Pour (INIT: i←0 ; TEST : i<joueur ; MAJ : i++) FAIRE
        SI PfTableau[i][7] =/=0 ALORS
                minute ←(PfTableau[i][7]/1000)/60
                seconde←(PfTableau[i][7]/1000)%60
                AFFICHER "Le participant n°"(PfTableau[i][0])" a fait un temps de "(minute)"m et "(seconde)"s."
        FIN SI
FIN POUR
FIN
```

4.Jeux d'essai

pfTab	Résultat attendu
{1, 1, 100000, 8, 1, 80000, 52, 240000}, {2, 1, 120000, 6, 1, 90000, 4, 220000}, {3, 1, 94000, 2, 0, -, -, 0}	Le participant n°1 a fait un temps de 4m et 0s. Le participant n°2 a fait un temps de 3m et 40s. (Rien ne s'affiche pour le participant n°3)

```
Le participant n°1 a fait un temps de 4m et 0s.
Le participant n°2 a fait un temps de 3m et 40s.
```

Explication :

Le principe est le même que le sous programme précédent, seulement on ne prend pas en compte les valeurs de PfTableau[i][7]==0, et on fait un léger calcul pour transformer les millisecondes en minutes et secondes afin de rendre le résultat plus lisible.

5/Bilan

-Difficultés et Réussites

-J’ai eu quelques difficultés à adapter mes codes à ceux de mes camarades. Je me suis principalement occupé de petites envergures, mais en grand nombres, la difficulté principale était donc de rendre ces codes compatibles avec ceux de mes collègues.

J’ai cependant bien réussit à tout faire, j’ai également eu le travail à la fin de rassembler tous les codes (qui avaient été fait séparément) et de modifier légèrement le tout et le tester jusqu’à ce que ce dernier fonctionne

-Ce que j’ai Appris :

-J’ai appris à décomposer un problème d’envergure moyenne en nombreux petits problèmes, je me suis en effet occupé dès le début de réduire l’important programme demandé en nombreux plus petits programmes que nous nous sommes ensuite répartis.

-J’ai appris et su maîtriser la manipulation d’un tableau à plusieurs dimensions. Comment le parcourir, savoir récupérer et manipuler des données à l’intérieur de ce dernier.

-J’ai appris à faire une documentation complète pour présenter un code. Ayant fait le code pour le main, j’ai dû prendre du temps à lire et comprendre les codes de mes collègues, puis à les décrire pour expliquer le fonctionnement complet du main qui appelle tous ces sous-programmes.

Grâce à ça mon code est plus compréhensible, il est plus facile à reprendre et à modifier par quelqu’un d’autre.

-Ma progression dans les compétences critiques :

-J’estime que ce projet joue sur plusieurs des compétences importantes requises dans ce domaine :

-C4/Réaliser et développer une application : Je me suis amélioré en faisant l’expérience du développement d’un outil plus complexe que ce que j’ai l’habitude de faire, ce fut un défi plus intéressant que les travaux demandés d’habitude

-C3/Optimisation des applications informatiques : J'ai dû mettre ensemble tous les codes et les faire fonctionner, ce faisant j'ai passé un certain temps à étudier les codes de mes collègues et à apporter de légères modifications pour améliorer ces derniers et les permettre de fonctionner les uns avec les autres.

-C5/Conduire un Projet : J'estime avoir eu une certaine importance dans la direction et l'avancement du projet, ayant produit un croquis des sous programmes et ayant communiqué avec mes camarades pour la répartition, selon quel membre se sentait plus à l'aise avec quoi. J'estime cependant qu'il n'y a pas eu de «rôle de chef» donné dans notre groupe, et que tout c'est fait plutôt à travers de la communication entre chaque membre selon les besoins de chacun pour atteindre un but commun.

-C6/Travailler dans une équipe informatique : cette compétence fait suite à ce que je disais dans la précédente. Il y a eu une bonne ambiance au sein du groupe et une bonne communication, tout le monde a fait sa part du travail en tenant les autres au courant de leur avancée. Les tâches ont été correctement réparties et ont été accomplis avec efficacité.

-Améliorations à apporter :

J'ai passé beaucoup de temps à raffiner, peaufiner le programme. C'est pour cela que je suis sûr qu'il est possible de penser encore à des cas que nous n'avons pas prévu, ou encore d'améliorer l'affichage pour le rendre plus lisible ou plus précis.

Il est sûrement possible aussi d'employer d'autres méthodes que celles utilisaient pour rendre le code plus léger à la lecture (humaine) ou à l'utilisation (processeur).