

```
// Phase 2 DA
// CONSTANTES
```

```
MAX_TEMPS = 601
MAX_LINES_TRAINSET 330
// 330 car j'ai pris de manière aléatoire 90% des subs parmi les 24
// ce qui me fait 22 lignes par file pour le trainSet donc au total 22*15 = 330
MAX_MOVEMENTS = 6
MAX_CHAR_NAME_MOVEMENTS 50
MAX_CHAR_LINE 50000 // Un nombre grand pour garantir que aucune donnée ne soit perdu lors de l'extractNumbersCSVion des doubles
TRAINSET_NAME "trainSet.csv"
PATTERN_NAME "pattern.csv"
```

```
struct subStats
{
char fileName[MAX_CHAR_NAME_MOVEMENTS]
double stats [MAX_TEMPS ]
}
```

```

Main
*
statsSub = déclaration d'un tab 2D de MAX_LINES_TRAINSET * MAX_TEMPS de type subStats
averageMovements = déclaration d'un tab 2D de MAX_MOVEMENTS * MAX_TEMPS de type double
movementsNames[0] = "dws"
...
movementsNames[5] = "wlk"

// Initialiser les tableaux à 0
i = 0
while (i < MAX_LINES_TRAINSET)
    ↓ statsSub[i].stats
    initTab
    ↓ statsSub[i].stats
    i++

i = 0
while (i < MAX_MOVEMENTS )
    ↓ averageMovements[i]
    initTab
    ↓ averageMovements[i]
    i++

// Extraire les données du trainSet dans le tableau statsSub
    ↓ statsSub
    extractNumbersCSV
    ↓ statsSub

// Utiliser la notion du bloc logique pour traiter les données correctement
iline = 0
indexMovement = 0
marker = ARRAY(MAX_TEMPS) // tableau de suivi pour le nbFilePerMovement
while(iline < MAX_LINES_TRAINSET)

    // Inititaliser chaque cellule de marker à 0
    ↓ marker
    initTab
    ↓ marker

    currentMovement = movementsNames[indexMovement]
    while(iline < MAX_LINES_TRAINSET AND currentMovement statsSub[iline].fileName)
        iTemps = 0
        while(iTemps < MAX_TEMPS)
            averageMovements[indexMovement][iTemp] += statsSub[iline].stats[iTemps]
            // Marque les cases remplis dans le csv pour ne pas compter les cases vide de certaines lignes
            if(statsSub[iline].stats[iTemps] ≠ 0)
                marker[iTemp] += 1
            iTemps++
        iline++
        iTemps = 0
        while(iTemps < MAX_TEMPS)
            averageMovements[indexMovement][iTemp] /= marker[iTemps]
            iTemps++
        indexMovement++

    ↓ averageMovements, movementsNames
    printCSVAverages
Sortir "Traitement terminé"
```

```

    ↓ tab
    initTab
    ↓ tab
    *
```

```

i = 0
while (i < MAX_TEMPS)
  tab[i] = 0

```

```

extractNumbersCSV ↓ numbers
extractNumbersCSV ↓ numbers

```

```

*
file = ouvrir le dossier TRAINSET_NAME
if (file == NULL)
  Sortir "Erreur lors de l'ouverture du fichier"
else
  line = allocation dynamique pour un tab de MAX_CHAR_LINE
  if(line == NULL)
    Sortir "Erreur d'allocation mémoire"
  else
    lineNumber = 0
    // Ignorer la premier ligne (ligne d'en-tête)
    fgets(line,MAX_CHAR_LINE,file)

    while(lineNumber < MAX_LINES_TRAINSET)

      Ouvrir la ligne du csv en cours
      // fgets(line,MAX_CHAR_LINE,file)

      nextToken = NULL
      token = strtok_s(line,",",&nextToken)

      // Stocker le nom du file en cours (Ex : dws_1)
      numbers[lineNumber].fileName = token
      iTemps = 0
      while(iTemps < MAX_TEMPS + 3 AND token != NULL)
        // Puisque le csv est composé pour chaque ligne d'un nom de mouvement, d'un index et un genre,
        // il suffit d'ignorer les colonnes qui nous intéressent pas
        if(iTemps ≥ 3)
          numbers[lineNumber].stats[iTemps-3] = token

        iTemps++
        token = strtok_s(NULL,",",&nextToken)

      lineNumber++

  libérer l'espace mémoire de line
  fermer le fichier file

```

```

printCSVAverages ↓ data,movementsName

```

```

*
file = ouvrir le fichier PATTERN_NAME
if(file == NULL)
  Sortir "Erreur lors de l'ouverture du fichier"
else
  Sortir le titre "Mouvement"
  iTemps = 0
  while(iTemps < MAX_TEMPS)
    Sortir la colonnes ",Vacc %d", iTemps
    iTemps++

  Sortir un passage à la ligne suivant "\n"

  iMovement = 0
  while(iMovement < MAX_MOVEMENTS)
    Sortir le nom du mouvement en cours avec "%s" movementsNames[iMovement]
    iTemps = 0
    while(iTemps < MAX_TEMPS)
      Sortir la valeur de la cellule "%f" data[iMovement][iTemps]
      iTemps++

    Sortir un passage à la ligne suivant "\n"
    iMovement++

```