

OC INFORMATIQUE

Le contexte commun à tous les exercices de cet examen est le TENNIS. Tous les exercices ont le même poids.

Rappels : Un match est gagné en 2 ou 3 sets gagnant selon le tournoi considéré. Un set est gagné lorsqu'un joueur a gagné 6 jeux avec deux jeux au moins d'écart avec son adversaire : 6-4, 6-3, 6-2, 6-1 ou 6-0. Les autres scores permettant de gagner un set sont 7-6 et 7-5. On considère donc ici que, s'il le faut, tous les sets se jouent "au tie-break". Chaque joueur possède un "classement ATP" qui est son rang dans la hiérarchie mondiale des joueurs professionnels.

Remarque : L'examen est fait à l'ordinateur avec des fichiers de base proposés. Vous êtes libres de ne pas utiliser ou de n'utiliser que partiellement le contenu de ces fichiers.

Exercice 1

Jeu, set et match

Compléter le fichier de base afin que soient réalisés les buts suivants.

- 1) Ecrire l'**expression régulière** la plus simple possible qui permet de valider des résultats de sets terminés. ("5-3" résultat d'un set en cours ne doit donc pas être accepté)
- 2) Ecrire le **code de la fonction *simuleSet(r1,r2)*** qui permet de simuler un set joué entre 2 joueurs (j1 et j2) de classement ATP r1 et r2. Chaque jeu du set est gagné par j1 ou j2 selon le "modèle" suivant : j1 a une probabilité $p1 = \frac{r1}{r1+r2}$ de gagner, alors que j2 a une probabilité $p2 = 1 - p1 = \frac{r2}{r1+r2}$ de gagner.
Le programme simulera donc le tirage d'un nombre aléatoire pour déterminer, pour chaque jeu, si c'est j1 ou j2 qui l'emporte.
La fonction *simuleSet* doit renvoyer le résultat du set sous la forme d'un tableau à 2 valeurs, par exemple [6,4].
Exemple : *simuleSet(1,9)* simule un set entre le joueur de rang 1 (le meilleur) et de celui de rang 9. Ainsi $p1 = \frac{9}{1+9} = 0.9$ (et $p2 = 0.1$). On tire alors un nombre aléatoire entre 0 et 1: si ce nombre est plus petit que 0.9 on attribuera le jeu au joueur 1, sinon au joueur 2. Etc... jusqu'à la fin du jeu.
- 3) Considérons un tournoi lors duquel les matchs se jouent en 2 sets gagnants. Ainsi des scores possibles de tels matchs sont, par exemple : "6-4,6-2", "6-7,6-2,5-7", "7-6,7-5".

Ecrire le **code du constructeur *Score(txt,j1,j2)*** qui a en argument *txt* une chaîne de caractères qui contient le score d'un match sous la forme des exemples ci-dessus, ainsi que *j1* et *j2* les noms des joueurs.

L'objet *Score* doit contenir les attributs : *nom1* (le nom du 1er joueur), *nom2* (le nom du 2ème joueur) et *sets* (un tableau bidimensionnel qui contient les résultats par sets. Par exemple : [[6,4],[6,2]]).

Les méthodes de l'objet *Score* sont : *jeux(k)* (renvoie le nombre de jeux gagnés par le joueur k (k=1 ou k=2)) et *vainq()* (renvoie le nom du vainqueur du match).

OC INFORMATIQUE

Exercice 2

Défilement, heure et prochaine finale

Compléter le fichier de base afin que soient réalisés les buts suivants.

- 1) On souhaite qu'une image défile de gauche à droite dans un canevas de manière continue (elle revient à gauche après être sortie du cadre). Le défilement sera de 2 pixels sur la droite chaque 30 millisecondes. Le défilement doit commencer à l'ouverture de la page. Lors d'un clic sur le canvas l'image devra être repositionnée tout à gauche.
L'image à placer initialement est "atp1.jpg", qu'on redimensionnera de taille 100px sur 100px.
Toutes les 2 secondes on souhaite que l'image soit modifiée en prenant l'image suivante dans la liste d'images. Cette liste contient les noms de trois fichiers images : "atp1.jpg", "atp2.jpg" et "atp3.jpg", qu'on parcourra donc "en boucle" en changeant toutes les 2 secondes.
- 2) On souhaite afficher l'heure actuelle (heure/minute/seconde), actualisée chaque seconde, dans la *div* nommée *heure*.
- 3) On souhaite que la *div* nommée *annonce* affiche, à l'ouverture de la page, le nombre de jours jusqu'à une prochaine finale d'un tournoi du circuit ATP ainsi que le ou les lieux de cette/ces finale(s).



Attention : il n'est pas autorisé d'utiliser la méthode Sort.

Le tableau *x* contient les noms et dates des finales des tournois du circuit ATP.
En voici un exemple partiel :

```
<script>
    var x= [ {lieu:"Roland-Garros",date:"2016-06-05"},
              {lieu:"Australian Open",date:"2017-01-29"},
              {lieu:"US Open",date:"2016-09-11"},
              {lieu:"Wimbeldon",date:"2016-07-10"}, ... ]
</script>
```

OC INFORMATIQUE

Exercice 3

Bases de données ATP

Les solutions de cet exercice peuvent être rédigées avec un traitement de texte ou en version manuscrite.

Une base de données contient les tables **joueurs** et **finale**s. Ces tables contiennent des informations respectivement sur les joueurs du circuit ATP et sur les finales qui ont déjà été disputées.

La table **joueurs** a les attributs : *idJ* (la clé primaire auto-incrémentée), *nom*, *rang*, *nation*, *taille*.

La table **finale**s a les attributs : *idF* (la clé primaire auto-incrémentée), *lieu*, *pays*, *date*, *idJ1* (clé étrangère), *idJ2* (clé étrangère), *nbSpec*.

Important : Le joueur qui a gagné la finale est toujours *idJ1*.

Les champs *pays* et *nation* contiennent les abréviations des noms des pays (F,CH,D,ESP,USA,...).

nbSpec est le nombre de spectateurs ayant assistés à la finale.

Aide : Pour utiliser deux fois une table dans une requête il faut lui donner deux alias différents.

Par exemple la requête

SELECT j1.nom , j2.nom from joueurs j1, joueurs j2 where j1.rang=1 and j2.rang=2
permet d'afficher dans deux colonnes le nom du joueur de rang 1 (colonne 1) et de celui de rang 2 (colonne 2).

Ecrire les requêtes SQL qui satisfont les demandes suivantes :

- 1) Ajouter un nouveau joueur dans la table des joueurs : il est Français, mesure 189 cm, s'appelle "Henri" et son rang est le 989.
- 2) Suite à l'arrêt de compétition du joueur de rang 101, tous les joueurs classés après lui voient leur rang amélioré de 1. Ecrire les requêtes qui permettent de "sortir" ce joueur de la table et de faire les modifications des rangs nécessaires.
- 3) Afficher les tailles minimales et maximales des joueurs, par nationalité.
- 4) Afficher les noms et tailles des joueurs dont la taille est supérieure à la moyenne des tailles de l'ensemble des joueurs.
- 5) Afficher le rang le plus grand d'un vainqueur d'une finale.
- 6) Afficher les dates des finales gagnées par un Suisse (CH), la plus récente en premier.
- 7) Afficher le nombre de finales gagnée par les joueurs du top 10 ($\text{rang} \leq 10$) en 2015.
- 8) Afficher le nombre de finales avec (au moins) un des joueurs finalistes classé après le rang ATP 15 ($\text{rang} > 15$).

OC INFORMATIQUE

Exercice 4

Canvas

Cet exercice permet de tester votre capacité à comprendre et vous approprier le code d'un programme pour le faire "évoluer".

Le fichier mis à disposition permet de représenter les quarts de finale (Q1/Q2/Q3/Q4), les demi-finales (D1/D2) et la finale (F) d'un tournoi ATP par un arbre binaire dans un canevas. Ces 7 matchs sont représentés, classiquement, par des rectangles (des "Eti"quettes).

Deux tableaux contiennent les informations utiles : **listeEti** et **listeMatches**. L'indice 0 n'est pas utilisé. Pour les indices 1 à 7, le même indice identifie l'étiquette et le match. Ainsi l'élément d'indice k de listeEti est l'étiquette qui correspond au match d'indice k de listeMatches.

Modifier/ajouter ce qu'il faut dans le code donné afin de réaliser les buts suivants.

- 1) Ecrire le code de la méthode *isIn(cx,cy)* de l'objet Eti. Cette méthode renvoie *true* si le point $(cx;cy)$ du canevas est dans l'étiquette, et *false* sinon.
- 2) Ecrire le code de la fonction *segment(x1, y1, x2, y2)* qui permet de tracer un segment entre les points $(x1; y1)$ et $(x2; y2)$ du canvas.

- 3) Lors d'un clic sur l'une des étiquettes du canvas :
 - afficher le résultat du match (sous la forme d'une balise < table >) dans la div nommée *resultat*.
Ecrire le nom du joueur vainqueur en gras.
Aide : la méthode ".bold()" de l'objet String permet la mise en gras de texte de manière aisée.

RESULTAT DU MATCH CHOISI

Federer 7 6 6
Berdych 6 2 4

- l'étiquette cliquée, et elle seule, doit être colorée.

- 4) Lorsqu'un joueur est sélectionné dans le menu déroulant :
 - colorer les étiquettes des matchs joués par ce joueur.
 - déterminer le nombre de matchs gagnés par ce joueur et en conclure et afficher ("Quart de finaliste", "Demi finaliste", "Finaliste" ou "Vainqueur") selon le stade atteint par ce joueur dans le tournoi.
 - afficher le nombre de sets joués par ce joueur.
 - afficher le nombre de jeux disputés par ce joueur.

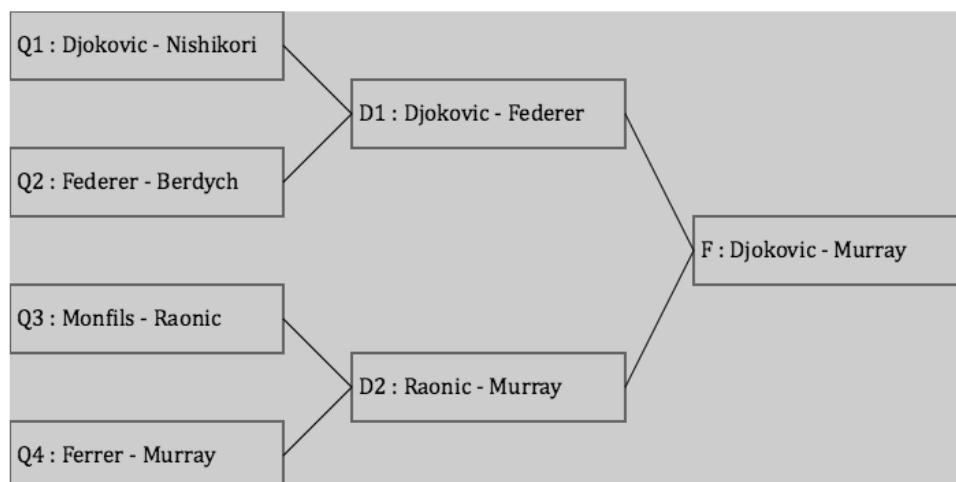
Ces informations sont à afficher l'une sous l'autre dans la *div* nommée *infos*.

CHOISIR UN JOUEUR : Federer

Demi finaliste
Nb de sets : 7
Nb de jeux : 64

OC INFORMATIQUE

Illustration – Exercice 4



OC INFORMATIQUE

Solutions- Exercice 1

1) `/^(6-[0-4]|[0-4]-6|7-6|7-5|6-7|5-7)$/`

```
2) function simuleSet(r1,r2){
    x=r2/(r1+r2)
    sc1=0
    sc2=0
    fini=false
    while(!fini){
        alea=Math.random()
        if (alea<x) {
            sc2++
        } else {sc1++}
        if ((sc1==6 && sc2<5) || (sc2==6 && sc1<5) || (sc1==7) || (sc2==7)) {
            fini=true
        }
    }
    res=[sc1,sc2]
    win=1*(sc1>sc2)+2*(sc2>sc1)
    return res
}
```

```
3) function Score(txt,j1,j2) {
    this.nom1=j1
    this.nom2=j2
    this.sets=[[txt[0],txt[2]], [txt[4],txt[6]], [txt[7],txt[9]]]
    this.jeux=function(k){
        if (k==1) {return
        parseInt(this.sets[0][0])+parseInt(this.sets[1][0])+parseInt(this.sets[2][0])}
        else {return
        parseInt(this.sets[0][1])+parseInt(this.sets[1][1])+parseInt(this.sets[2][1])}
    }
    this.vainq=function(){
        s1=(this.sets[0][0]>this.sets[0][1])+(this.sets[1][0]>this.sets[1][1])+(this.sets[2][0]>this.sets[2][1])

        s2=(this.sets[0][0]<this.sets[0][1])+(this.sets[1][0]<this.sets[1][1])+(this.sets[2][0]<this.sets[2][1])
        return 1*s1+2*s2
    }
}
```

OC INFORMATIQUE

Solutions- Exercice 2

```
<body onload="majheure()">
<canvas onclick="k=0" id="myCanvas" width="400" height="100" style="border:1px
solid #000000;"> </canvas>
</br> <div id="heure"> </div> </br> <div id="annonce"> </div>
<script>
```

// 1) Defilement

```
var listeImg=["atp0.jpg","atp1.jpg","atp2.jpg"]
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
var noImg=0
var my_img=new Image()
my_img.src=listeImg[0]
ctx.drawImage(my_img,0,-100,100,100)
var t1=setInterval("reset()",30)
var k=0
var
t2=setInterval("noImg++;if(noImg>2){noImg=0};my_img.src=listeImg[noImg]",2000)

function reset(){
  ctx.clearRect(-100,0,500,100)
  ctx.drawImage(my_img,k,0,100,100)
  k=k+2
  if (k>400) {k=-100}
}
```

// 2) heure

```
function majheure(){
  var now=new Date()
  document.getElementById("heure").innerHTML=now.toLocaleTimeString()
  setTimeout(majheure,1000)
}
```

OC INFORMATIQUE

// 3) Prochain tournoi

```
var x=[{lieu:"Paris",date:"2016-06-04"},{lieu:"Lisbone",date:"2016-06-05"},{lieu:"Séville",date:"2016-06-10"},{lieu:"Monaco",date:"2016-06-04"}]
function prochainTournoi(){
  k=0
  dmin=10000
  now=new Date()
  for (k in x){
    var t=x[k]
    dd=Math.floor((new Date(t.date)-now)/1000/60/60/24)
    if (dd>=0 && dd<dmin) {dmin=dd}
  }
  listeLieux=[]
  for (k in x){
    var t=x[k]
    dd=Math.floor((new Date(t.date)-now)/1000/60/60/24)
    if (dd==dmin) {listeLieux.push(t.lieu)}
  }
  document.getElementById("annonce").innerHTML=" J - " + dmin + " : "
+listeLieux.join(" et ")
}
</script>
</body>
```


OC INFORMATIQUE

Solutions- Exercice 3

Ajouter un nouveau joueur dans la table des joueurs : il est Français, mesure 189 cm, s'appelle "Henri" et son rang est le 989.

```
INSERT INTO `joueurs`(`idJ`, `nom`, `rang`, `nation`, `taille`) VALUES  
(0,'Henri',989,'F',189)
```

Suite à l'arrêt de compétition du joueur de rang 101, tous les joueurs classés après lui voient leur rang amélioré de 1. Ecrire les requêtes qui permettent de "sortir" ce joueur de la table et de faire les modifications des rangs nécessaires.

```
DELETE FROM `joueurs` WHERE rang=101  
UPDATE `joueurs` SET rang=rang-1 where rang>101
```

Afficher les tailles minimales et maximales des joueurs, par nationalité.

```
SELECT min(taille) as minimum, max(taille) as maximum, nation FROM `joueurs` group  
by nation
```

Afficher les noms et tailles des joueurs dont la taille est supérieure à la moyenne des tailles de l'ensemble des joueurs.

```
SELECT nom,taille FROM `joueurs` WHERE taille > (select avg(taille) from joueurs)
```

Afficher le rang le plus grand d'un vainqueur d'une finale.

```
select max(rang) from joueurs where idJ in (select idJ1 from finales)
```

Afficher les dates des finales gagnées par un Suisse (CH), la plus récente en premier.

```
SELECT distinct date from finales where idJ1 in (select idJ from joueurs where  
nation='CH') order by date DESC;
```

Afficher le nombre de finales gagnée par les joueurs du top 10 (rang<=10) en 2015.

```
select count(distinct idF) from finales, joueurs where idJ1=idJ and year(date)=2015 and  
rang<=10;
```

Afficher le nombre de finales avec (au moins) un des joueurs finalistes classé après le rang ATP 15 (rang > 15).

```
select count(distinct idF) from finales, joueurs j1, joueurs j2 where idJ1=j1.idJ and  
idJ2=j2.idJ and (j1.rang>15 or j2.rang>15);
```

Afficher les noms des deux finalistes de toutes les finales différentes disputées.

```
SELECT j1.nom nom1, j2.nom nom2 from joueurs j1, joueurs j2 where j1.idJ in (select  
idJ1 from finales where j2.idJ=idJ2);
```

OC INFORMATIQUE

Afficher le nombre de finales gagnées par le plus petit des deux finalistes.

```
select count(distinct idF) from finales, joueurs j1, joueurs j2 where idJ1=j1.idJ and  
idJ2=idJ2 and j1.taille<j2.taille;
```

Afficher la liste des pays dans l'ordre décroissant du nombre de spectateurs total.

```
select pays, sum(nbspec) as Spectateurs_total from finales group by pays order by  
Spectateurs_total desc;
```

Afficher le nombre de spectateurs moyen des finales gagnées par le joueur "Federer".

```
ELECT avg(nbSpec) FROM joueurs, finales WHERE idJ1 = (select idj from joueurs where  
nom="Federer")
```

Afficher le nombre de fois que des joueurs de même nationalité se sont affrontés en finale.

```
SELECT j1.nom nom1, j2.nom nom2 from joueurs j1, joueurs j2, finales where j1.idJ=idJ1  
and j2.idJ=idJ2 and j1.nation=j2.nation;
```

Afficher le nom du vainqueur de la finale et le nombre de spectateurs de la finale qui s'est jouée devant le plus grand nombre de spectateurs.

```
select nom, nbspec from joueurs, finales where idJ1=idJ and nbspec=(select  
max(nbspec) from finales)
```

OC INFORMATIQUE

Solutions- Exercice 4

```
this.isIn=function(cx,cy){
    return (cx>this.x)&&(cx<this.x+200)&&(cy>this.y)&&(cy<this.y+50)
}

function segment(x1,y1,x2,y2){
    ctx.beginPath();
    ctx.moveTo(x1,y1);
    ctx.lineTo(x2,y2);
    ctx.stroke();
    ctx.closePath();
}

function affiche(ex,ey){
    cx=ex-document.getElementById('monCanvas').offsetLeft
    cy=ey-document.getElementById('monCanvas').offsetTop
    kin=0
    for (k=1;k<=7;k++){
        if (listeEti[k].isIn(cx,cy)) {
            kin=k
        }
    }
    if (kin!=0) {
        init()
        listeEti[kin].plein()
        afficheResultat(kin)
    }
}

function afficheResultat(k){
    match=listeMatches[k]
    if (match.j1==match.win) {
        aux="<table><tr><td>"+match.j1.bold()+"</td><td>"+match.res1.join("
")+ "</td></tr>"
        aux+="<tr><td>"+match.j2+"</td><td>"+match.res2.join("
")+ "</td></tr></table>"
    }
    else{
        aux="<table><tr><td>"+match.j1+"</td><td>"+match.res1.join("
")+ "</td></tr>"
        aux+="<tr><td>"+match.j2.bold()+"</td><td>"+match.res2.join("
")+ "</td></tr></table>"
    }
    document.getElementById("resultats").innerHTML=aux
}
```

OC INFORMATIQUE

```
function informations(joueur){
    init()
    matchG=0
    nbSets=0
    nbJeux=0
    for (k=1;k<=7;k++) {
        if (listeMatches[k].j1==joueur || listeMatches[k].j2==joueur) {
            listeEti[k].plein()
            nbSets+=listeMatches[k].res1.length
            for (m in listeMatches[k].res1){nbJeux+=listeMatches[k].res1[m]}
            for (n in listeMatches[k].res2){nbJeux+=listeMatches[k].res2[n]}
        }
    }
    for (k=1;k<=7;k++){
        if (listeMatches[k].win==joueur){
            matchG++
        }
    }
    if (matchG==0) {aux="Quart de finaliste"}
    if (matchG==1) {aux="Demi finaliste"}
    if (matchG==2) {aux="Finaliste"}
    if (matchG==3) {aux="Vainqueur"}
    aux=aux+"</br>"+ "Nb de sets : "+nbSets
    aux=aux+"</br>"+ "Nb de jeux : "+nbJeux
    document.getElementById("infos").innerHTML=aux
}
```