# **Problème N° 1**

# *(durée 60 min, 20 points)*

***Réactions chimiques – Production de code.***

Le but de ce problème est de visualiser l’évolution au cours du temps des quantités des composés chimiques lors d’une réaction.

Un exemple de fonctionnement de cette application est disponible dans le fichier vidéo *resultat.mp4*.

Une réaction chimique est représentée par une équation :

a A + b B m C + n D

1. Ecrire le code de la fonction ***imprimeEquation()*** qui va récupérer les informations dans les champs *INPUT* des réactifs et des produits (les coefficients *a, b, m, n* et les composés chimiques A, B, C, D) et réintroduire l’équation chimique par *innerHTML* en ajoutant la balise *sub* et pour afficher correctement les indices des composés chimiques dans la *div* correspondante (*id* = "equ"), en utilisant le moins de balise *sub* que possible.

Par exemple, CH3CH2OH sera codé "CH<sub>3</sub>CH<sub>2</sub>OH" pour être affiché CH3CH2OH dans la page web.

Ne pas oublier de laisser l’affichage de la flèche indiquant le sens de la réaction. L’image de la flèche est dans le fichier *fleche.jpg*.

1. Ecrire le code de les fonctions ***reaction()*** et ***iteration()*** qui va simuler par itération la variation des quantités des composés pendant la réaction chimique au cours du temps.

La simulation se fait dans un *canvas* placé sous l’équation chimique en faisant évoluer un graphique en histogramme au cours du temps.

A chaque itération, les concentrations des réactifs diminuent en fonction de leurs coefficients et les concentrations des produits augmentent en fonction de leurs coefficients. Le calcul de la modification des concentrations CA, CB, CC et CD respectivement des composés A, B, C et D se fait donc de la façon suivante :

(CA)i = (CA)i-1 – *a*·*x* (CB)i = (CB)i-1 – *b*·*x*

(CC)i = (CC)i-1 + *m*·*x* (CD)i = (CD)i-1 + *n*·*x*

(CA)i, (CB)i, (CC)i et (CD)i correspondent aux concentrations à l’étape *i* et *x* au pas du calcul.

La vitesse de la simulation va dépendre du choix de *x*. Plus il sera petit et plus l’évolution de la réaction sera lente. *x* doit pouvoir être choisi par l’utilisateur en l’introduisant dans un champ *INPUT*.

La simulation doit s’interrompre lorsque la concentration CA ou CB est égale à zéro : A ou respectivement B sera le réactif limitant.

Il faudra imprimer dans une fenêtre surgissante lequel des deux réactifs est le réactif limitant ou si, par hasard, les deux sont complètement consommés.

# **Problème N° 2**

# *(durée 60 min, 20 points)*

***Shazam moléculaire – Corrections et optimisation d’un code***

L’application Shazam permet de reconnaître un morceau de musique après en avoir écouté quelques dizaines de secondes. En fait, cette application fait une mesure de similarité.

L’idée de la mesure de la similarité sera adaptée ici pour programmer une comparaison de la réactivité chimique (la similarité) entre deux molécules. L’indice de similarité utilisé pour comparer deux molécules A et B s’appelle le coefficient de Tanimoto *T* :



*N11* : nombre de cases où l’on trouve 1 à la fois chez A et chez B.

*N10* : nombre de cases où l’on trouve 1 chez A et 0 chez B.

*N01* : nombre de cases où l’on trouve 0 chez A et 1 chez B.

Exemple :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A : | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| B : | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |

Cela donne : 

1. S’approprier le code du fichier *probleme2.html* et corriger les fautes que ce programme comporte :

* Le deuxième menu déroulant ne fonctionne pas correctement : il modifie la mauvaise molécule. A corriger.
* Il faut automatiser le remplissage des champs *INPUT* correspondants lorsqu’on sélectionne la molécule du menu déroulant avec les données ci-dessous :

Diclofenac : *vecteur* = [1, 1, 1, 0, 1, 0, 0, 0]

Lumiracoxib : *vecteur* = [1, 1, 1, 1, 1, 0, 0, 0]

Erlotinib : *vecteur* = [0, 1, 1, 1, 1, 1, 1, 1]

Gefitinib : *vecteur* = [0, 1, 0, 0, 1, 1, 1, 1]

* Le bouton « Calcul de T » disparaît lorsqu’on lance ledit calcul. Ce n’est pas une bonne idée. Il faut l’empêcher de disparaître.
* Il faut changer les mots « mol1 » et « mol2 » dans la réponse "Similarité entre mol1 et mol2 : T = " + *T* envoyée par *innerHTML* dans le paragraphe de résultat (*id = "res"*) par les noms des deux molécules sélectionnées dans les menus déroulants.

1. Optimiser le code du fichier *probleme2.html* :

* Simplifier les 16 lignes de récupération des valeurs des champs *INPUT* en créant un code le plus court possible.
* Simplifier la structure des tests permettant le calcul du coefficient de Tanimoto *T*.
* Programmer une fonction ***init()*** qui sera lancée lors du chargement de la page web et qui va créer automatiquement la table complète où seront affichées : les menus déroulants, les molécules, les fragments de molécules, ainsi que les 16 champs *INPUT*.

La table est à insérer entre *<div id= "table">* et *</div>* lors du chargement de la page.

A disposition pour ce problème, tous les fichiers *JPEG* dans le dossier *imgProb2*.

# **Problème N° 3**

# *(durée 40 min, 20 points)*

***Base de données chimique – Schéma Entité/Association et requêtes SQL***

Il faut concevoir une base de données pour des composés organiques. Pour gérer cette BD, il faudra déjà les tables MOLECULES, ATOMES, FONCTIONS et REACTIVITE. Ces tables contiendront des informations chimiques permettant de connaître la réactivité des molécules enregistrées.

Les tables ont les attributs suivants :

* MOLECULES : ***idMol*** (clé primaire auto-incrémentée), *nomMol*, *formuleMol*, *masseMolaire*, *idReac* (clé étrangère).
* ATOMES : ***idZ*** (clé primaire correspondant au numéro atomique), *nomAtome*, *symboleChim*, *masseAtomique*.
* FONCTIONS : ***idFonc*** (clé primaire auto-incrémentée), *nomFonc*, *formuleFonc*.
* REACTIVITE : ***idReac*** (clé primaire auto-incrémentée), *type*.

Il faudra aussi 2 tables d’association, entre MOLECULES et ATOMES et entre MOLECULES et FONCTIONS, avec les attributs suivants :

* moleculesXatomes : *idMol* (clé étrangère), *idZ* (clé étrangère), *nombreAtomes*.

(La clé primaire ***idMolidZ*** est la combinaison des deux clés étrangères).

* moleculesXfonctions : *idMol* (clé étrangère), *idFonc* (clé étrangère).

(La clé primaire ***idMolidFonc*** est la combinaison des deux clés étrangères).

Les tables avec leurs données sont imprimées dans l’annexe 1 et également disponibles dans le fichier Excel *probleme3.xlsx*.

1. Dessiner le schéma entité/association complet, en utilisant les tables en annexe. Ne pas oublier les cardinalités. **Ce dessin se fait sur papier.**

Pour la suite du problème, il faudra enregistrer les réponses dans un fichier Word intitulé *probleme3.docx*.

1. Ecrire les requêtes SQL pour satisfaire les demandes ci-dessous.

Lors de la rédaction de chaque requête, il ne faut en aucun cas utiliser les données imprimées dans l’annexe 1.

* + 1. Extraire de la table ATOMES tous les atomes dont la masse atomique est plus petite que 16 et afficher toutes les données avec les atomes triés par ordre alphabétique de leurs noms.
    2. Dans la table MOLECULES, trouver le nom et la formule chimique de la molécule avec une masse molaire de 58.08 g/mol.
    3. Ajouter dans la base de données la molécule d’éthylamine (C2H7N, 45.09 g/mol, avec la réactivité = "base" et la fonction = "amine").

*La question 3) nécessite 2 requêtes pour insérer toutes les données (une requête dans MOLECULES et une dans moleculesXfonctions).*

1. Ecrire le résultat qui sera affiché par les requêtes ci-dessous :
   * 1. select \* from MOLECULES where idMol in (select idMol from moleculesXfonctions where idFonc in (select idFonc from FONCTIONS where nomFonc = "cétone"));
     2. select nomMol, masseMol from MOLECULES where idReac = (select idReac from REACTIVITE where type = "nucléophile") order by masseMol;
2. Dans la table MOLECULES, il y a 2 attributs de trop (2 colonnes inutiles). Lesquelles ? Expliquer pourquoi.

L’explication sera enregistrée sous la table dans la feuille Excel et les colonnes supprimées surlignées en jaune.

# **Problème N° 4**

# *(durée 20 min, 10 points)*

***Circuits logiques – Utilisation de Logisim***

A l’aide de l’application Logisim, résoudre les 2 questions ci-dessous :

1. Dessiner le circuit logique d’un additionneur complet sur 2 bits.

Le résultat doit être sauvegardé sous le nom *add2bits.circ*.

Rédiger la table de vérité de ce circuit dans un fichier Excel enregistré sous le nom *add2bits.xlsx*.

1. Dessiner le circuit logique permettant d’afficher le vecteur de 8 bits pour la molécule sélectionnée (Diclofenac ou Gefitinib).

Le résultat doit être enregistré sous le nom *molecules.circ*.

Données :

Diclofenac : *vecteur* = [1, 1, 1, 0, 1, 0, 0, 0]

Gefitinib : *vecteur* = [0, 1, 0, 0, 1, 1, 1, 1]

**Annexe 1** *Tables et données du problème 3*

***Tables d’entités***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MOLECULES** | | | | |
| **idMol** | **nomMol** | **formuleMol** | **masseMol** | **idReac** |
| 1 | acide acétique | C2H4O2 | 60.05 | 1 |
| 2 | benzène | C6H6 | 78.11 | 8 |
| 3 | acétone | C3H6O | 58.08 | 7 |
| 4 | hexanol | C6H14O | 102.18 | 5 |
| 5 | pent-4-én-2-ol | C5H10O | 86.13 | 5 |
| 6 | 4-hydroxypentan-2-one | C5H10O2 | 102.13 | 5 |
| 7 | hexane-2,4-dione | C11H20O3 | 200.28 | 5 |
| 8 | chloroforme | CHCl3 | 119.37 | 7 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ATOMES** | | | |
| **idZ** | **nomAtome** | **symboleChim** | **masseAtomique** |
| 1 | hydrogène | H | 1.01 |
| 2 | hélium | He | 4 |
| 6 | carbone | C | 12.01 |
| 7 | azote | N | 14 |
| 8 | oxygène | O | 16 |
| 17 | chlore | Cl | 35.45 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FONCTIONS** | | |  | **REACTIVITE** | |
| **idFonc** | **nomFonc** | **formuleFonc** |  | **idReac** | **type** |
| 1 | alcène | –C=C– |  | 1 | acide |
| 2 | acide | –COOH |  | 2 | base |
| 3 | alcool | –OH |  | 3 | oxydant |
| 4 | aldéhyde | –CHO |  | 4 | réducteur |
| 5 | cétone | –CO– |  | 5 | nucléophile |
| 6 | amine | –NH2 |  | 6 | électrophile |
| 7 | halogène | –X |  | 7 | solvant |
|  |  |  |  | 8 | aromatique |

***Tables d’associations***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **moleculesXatomes** | | |  | **moleculesXfonctions** | |
| **idMol** | **idZ** | **nombresAtomes** |  | **idMol** | **idFonc** |
| 1 | 6 | 2 |  | 1 | 2 |
| 1 | 1 | 4 |  | 2 | 1 |
| 1 | 8 | 2 |  | 3 | 5 |
| 2 | 6 | 6 |  | 4 | 3 |
| 2 | 1 | 6 |  | 5 | 1 |
| 3 | 6 | 3 |  | 5 | 3 |
| 3 | 1 | 6 |  | 6 | 3 |
| 3 | 8 | 1 |  | 6 | 5 |
| 4 | 6 | 6 |  | 7 | 5 |
| 4 | 1 | 14 |  | 8 | 7 |
| 4 | 8 | 1 |  |  |  |
| 5 | 6 | 5 |  |  |  |
| 5 | 1 | 10 |  |  |  |
| 5 | 8 | 1 |  |  |  |
| 6 | 6 | 5 |  |  |  |
| 6 | 1 | 10 |  |  |  |
| 6 | 8 | 2 |  |  |  |
| 7 | 6 | 11 |  |  |  |
| 7 | 1 | 20 |  |  |  |
| 7 | 8 | 3 |  |  |  |
| 8 | 6 | 1 |  |  |  |
| 8 | 1 | 1 |  |  |  |
| 8 | 17 | 3 |  |  |  |

**Annexe 2** *Formulaire*

**MATH : quelques attributs et méthodes de l’objet Math**

La syntaxe est "Math." suivi de la méthode.

|  |
| --- |
| *Méthode Description* |
| *PI* Renvoie le nombre **. |
| *abs(x)* Renvoie la valeur absolue de *x*. |
| *ceil(x)* Renvoie l’entier supérieur à *x*. |
| *exp(x)* Renvoie *ex*. |
| *floor(x)* Renvoie l’entier inférieur à *x*. |
| *max(x,y,z,...n)* Renvoie le maximum des nombres entrés en paramètres. |
| *min(x,y,z,...,n)* Renvoie le minimum des nombres entrés en paramètres. |
| *pow(x,y)* Renvoie *xy*. |
| *random()* Renvoie un nombre aléatoire dans l’intervalle [0 ; 1[. |
| *round(x)* Renvoie l’entier le plus proche de *x*. |
| *sqrt(x)* Renvoie la racine carrée de *x*. |

**NUMBER : quelques attributs et méthodes des nombres**

La syntaxe est "nomVariable." suivi de la méthode.

Exemple :

var num = 5.56789 ;

var n = num.toFixed(2) ;

Le résultat sera n = 5.57.

|  |
| --- |
| *Méthode Description* |
| *isInteger()* Teste si le nombre est un entier. Renvoie *true* ou *false*. |
| *isNaN()* Teste si ce n’est pas un nombre. Renvoie *true* ou *false*. |
| *toFixed(k)* Renvoie le nombre arrondi à *k* décimales (*k* = 0 est possible). |
| *toString()* Convertit le nombre en chaîne de caractères. |

**ARRAY : quelques méthodes utiles pour les tableaux**

|  |
| --- |
| *Méthode Description* |
| *x.concat(tab1, tab2[, tab3, ...])* Concatène *x* avec d’autres tableaux. *x* est modifié. |
| *x.pop()* Supprime le dernier élément du tableau et retourne sa valeur. |
| *x.push(valeur1[, valeur2, ...])* Ajoute un ou plusieurs éléments à la fin du tableau *x* et retourne le nombre d’éléments du tableau modifié. |
| *x.unshift(valeur1[, valeur2, ...])* Ajoute un ou plusieurs éléments au début du tableau *x* et retourne le nombre d’éléments du tableau modifié. |
| *x.reverse()* Inverse l’ordre des éléments du tableau. |
| *x.sort()* Trie les éléments du tableau. |
| *x.shift()* Supprime le premier élément du tableau et retourne sa valeur. |
| *x.slice(indiceDebut,indiceFin)* Renvoie le sous-tableau contenant les éléments de indiceDebut (inclus) à indiceFin (non inclus). *x* n’est pas modifié. |

**STRING : quelques méthodes utiles pour les chaînes de caractères**

|  |
| --- |
| *Méthode Description* |
| *x.charAt(index)* Lecture d’un caractère. |
| *x.charCodeAt(index)* Code Unicode d’un caractère (exemples : « A » : 65, « a » : 97). |
| *String.fromCharCode(x)* Caractère dont l’Unicode est le nombre *x*. |
| *x.toUpperCase()* Casse tout en majuscule. |
| *x.toLowerCase()* Casse tout en minuscule. |
| *x.toString()* Conversion du nombre x en String. |
| *x.split(car)* Conversion du String *x* à une Array. *car* est la variable de coupure. |
| *tab.join(car)* Conversion Array à String. |
| *txt1.concat(txt2,txt3,...)* Concaténation. |
| *x.replace(searchValue,newValue)* remplacement. |
| *x.slice(start,end)* Extraction : renvoie la sous-chaîne d’indice *start* (inclus) à *end* (non inclus). |
| *x.slice(start)* Extraction : renvoie la sous-chaîne d’indice *start* (inclus) à la fin. |
| *x.slice(-n)* Extraction : renvoie la sous-chaîne contenant les *n* derniers caractères. |

**EVENT : à propos des événements**

La syntaxe est "event." suivi de la méthode.

|  |
| --- |
| *Méthode Description* |
| *clientX, client Y, which* Coordonnée horizontale, verticale et n° du bouton de la souris (MouseEvent). |
| *which (ou charCode)* Code Unicode du caractère lors d’un événement lié au clavier (KeyboardEvent). |

**DATE : quelques méthodes de l’objet Date et l’instanciation**

|  |
| --- |
| *Méthode Description* |
| *new Date()* Génère la date (avec heures,...) actuelle. |
| *new Date(x,y,z)* Définit la date avec année *x*, mois *y* (0-11), jour *z* (1-31). |
| *new Date("YYYY-MM-DD")* Définit la date donnée en argument. |
| *getDay()* Indique le jour de la semaine : 0-6 (0 correspondant à dimanche). |
| *getDate()* Indique le jour du mois : 1-31. |
| *getMonth()* Indique le mois : 0-11. |
| *getFullYear()* Indique l’année sur quatre chiffres. |
| *getHours()* Indique l’heure : 0-23. |
| *getMinutes()* Indique les minutes : 0-59. |
| *getSeconds()* Indique les secondes : 0-59. |
| *getMilliseconds()* Indique les millisecondes : 0-999. |
| *getTime()* Indique le nombre de millisecondes entre la date du jour et le 01.01.1970. |
| *setFullYear(année,mois,jours,...)* Modifie l’année (quatre chiffres). *mois*, *jours*,... sont optionnels. |
| *setMonth(mois,jours,...)* Modifie le mois (0-11). *jours*,... sont optionnels. |
| *setDate(jours,h,...)* Modifie le jour (1-31). *h*,... sont optionnels. |
| *setHours(h,min,sec,millisec)* Modifie l’heure. *min*, *sec*, *millisec* sont optionnels. |
| *setMinutes(min,sec,millisec)* Modifie les minutes. *sec*, *millisec* sont optionnels. |
| *setSeconds(sec,millisec)* Modifie les secondes. *millisec* est optionnel. |
| *setMilliseconds(millisec)* Modifie les millisecondes. |
| *setTime(millisec)* Redéfinit une date en ajoutant *millisec* au 01.01.1970. *millisec* peut être négatif. |

**CANVAS : quelques méthodes graphiques**

Déclaration du canvas :

<body>

<canvas id="monCanvas" width="600" height="600" style=”border:1px solid black;”></canvas>

<script>

var c = document.getElementById(“monCanvas”);

var ctx = c.getContext(“2d”);

</script>

</body>

|  |
| --- |
| *Méthode Description* |
| *c.offsetLeft c.offsetTop* Position du canvas sur la page. |
| *ctx.beginPath()* Commence un chemin. |
| *ctx.closePath()* Lie le dernier point du chemin au premier. |
| *ctx.stroke()* Affiche le chemin. |
| *ctx.fill()* Remplit un chemin après l’avoir défini. |
| *ctx.moveTo(x,y)* Déplace le curseur en (x,y). |
| *ctx.lineTo(x,y)* Ajoute une ligne de la position du curseur jusqu’en (x,y). |
| *ctx.rect(x,y,width,height)* Dessine un rectangle dont le coin supérieur gauche est (x,y) de largeur width et de longueur height. |
| *ctx.fillRect(x,y,width,height)* Dessine un rectangle plein dont le coin supérieur gauche est (x,y) de largeur width et de longueur height. |
| *ctx.strokeRect(x,y,width,height)* Dessine le bord d’un rectangle dont le coin supérieur gauche est (x,y) de largeur width et de longueur height. |
| *ctx.clearRect(x,y,w,h)* Efface une zone rectangulaire. |
| *ctx.arcTo(x1,y1,x2,y2,r)* Trace un arc de cercle allant de (x1,y1) à (x2,y2) et de rayon r. |
| *ctx.arc(x,y,r,angleD,angleA,sens)* Trace un arc de cercle de centre (x,y), de rayon r, d’angle de depart et d’arrivée (en radians) donnés, et de sens anti-horaire si le dernier argument est absent ou vaut *true*. |
| *ctx.arc(x,y,r,0,2\*Math.PI)* Trace un cercle de centre (x,y) et de rayon r. |
| *ctx.strokeStyle = uneCouleur* Définit la couleur du chemin. |
| *ctx.fillStyle = uneCouleur* Définit la couleur de remplissage. |
| *ctx.lineWidth = 5* Définit l’épaisseur du chemin. |

|  |
| --- |
| *Méthode Description* |
| *ctx.globalAlpha = 0.6* Définit la transparence de la forme (0 : transparent à 1 : opaque). |
| *ctx.isPointInPath(x,y)* Renvoie *true* si le point (x,y) appartient au chemin actuel. |
| *ctx.font = "20px Arial"* Modifie la police. |
| *ctx.fillText("abc",x,y,maxWidth)* Ecrit le texte plein « abc » en (x,y), de largeur maximale donnée (optionnel). |
| *ctx.strokeText("abc",x,y,maxWidth)* Ecrit le texte vide « abc » en (x,y), de largeur maximale donnée (optionnel). |
| *ctx.measureText(texte)* Mesure d’un texte. |
| *ctx.translate(x,y)* Translation. |
| *ctx.scale(kx,ky)* Changement d’échelle. |
| *ctx.rotate(angle)* Rotation en radians. |
| *ctx.save()* Sauvegarde du canvas. |
| *ctx.restore()* Récupération du canvas. |

Insertion d’une image dans un canvas :

var monImage = new Image() ;

monImage.src = " ... “ ; // emplacement du fichier image.

var x = 10 ;

var y = 20 ;

ctx.drawImage(monImage,x,y) ;

L’instruction avec rognage rectangulaire et redimensionnement est :

ctx.drawImage(monImage,sx,sy,swidth,sheight,x,y,width,height) ;

Enfin, *ctx.clip()* permet des rognages pas nécessairement rectangulaires.

**Requêtes SQL**

|  |
| --- |
| *Expression Syntaxe* |
| AND / OR SELECT ... FROM ... WHERE ... AND/OR |
| AS (alias) SELECT ... AS ... FROM ... |
| BETWEEN SELECT ... FROM ...  WHERE ... BETWEEN ... AND |
| NOT EXISTS SELECT ... FROM ...  WHERE ... NOT EXISTS ... |
| DELETE DELETE FROM ... WHERE ... |
| DROP TABLE DROP TABLE ... |
| GROUP BY SELECT ... FROM ...  WHERE ... GROUP BY ... |
| IN SELECT ... FROM ... WHERE ... IN ( ... ) |
| NOT IN SELECT ... FROM ...  WHERE ... NOT IN ( ... ) |
| INSERT INTO INSERT INTO ... VALUES ... |
| LIKE / REGEXP SELECT ... FROM ... WHERE ... LIKE/-REGEXP ... |
| ORDER BY SELECT ... FROM ...  ORDER BY ... [ASC/DESC] |
| SELECT SELECT ... FROM ... |
| SELECT \* SELECT \* FROM ... |
| SELECT DISTINCT SELECT DISTINCT ... FROM ... |
| UPDATE UPDATE ... SET ... WHERE ... |
| WHERE SELECT ... FROM ... WHERE ... |

|  |
| --- |
| *Expression Description* |
| AVG() Moyenne. |
| COUNT() Nombre d’enregistrements. |
| MAX() / MIN() Plus grande / plus petite valeur. |
| SUM() Somme. |
| CONCAT() Concaténation. |
| ROUND() Arrondi d’une valeur numérique au nombre de décimales éventuellement spécifié. |
| NOW() Date d’aujourd’hui. |

|  |
| --- |
| *Expression Description* |
| YEAR() / MONTH() / DAY() Quelques autres fonctions pour les dates.  CURTIME() / CURDATE()  DATEDIFF(…,…) |
| LEFT(...,n) / RIGHT(...,n) LEN() Manipulation de texte.  UPPER() / LOWER() / LTRIM()  RTRIM() / SUBSTRING(str,pos,len) |