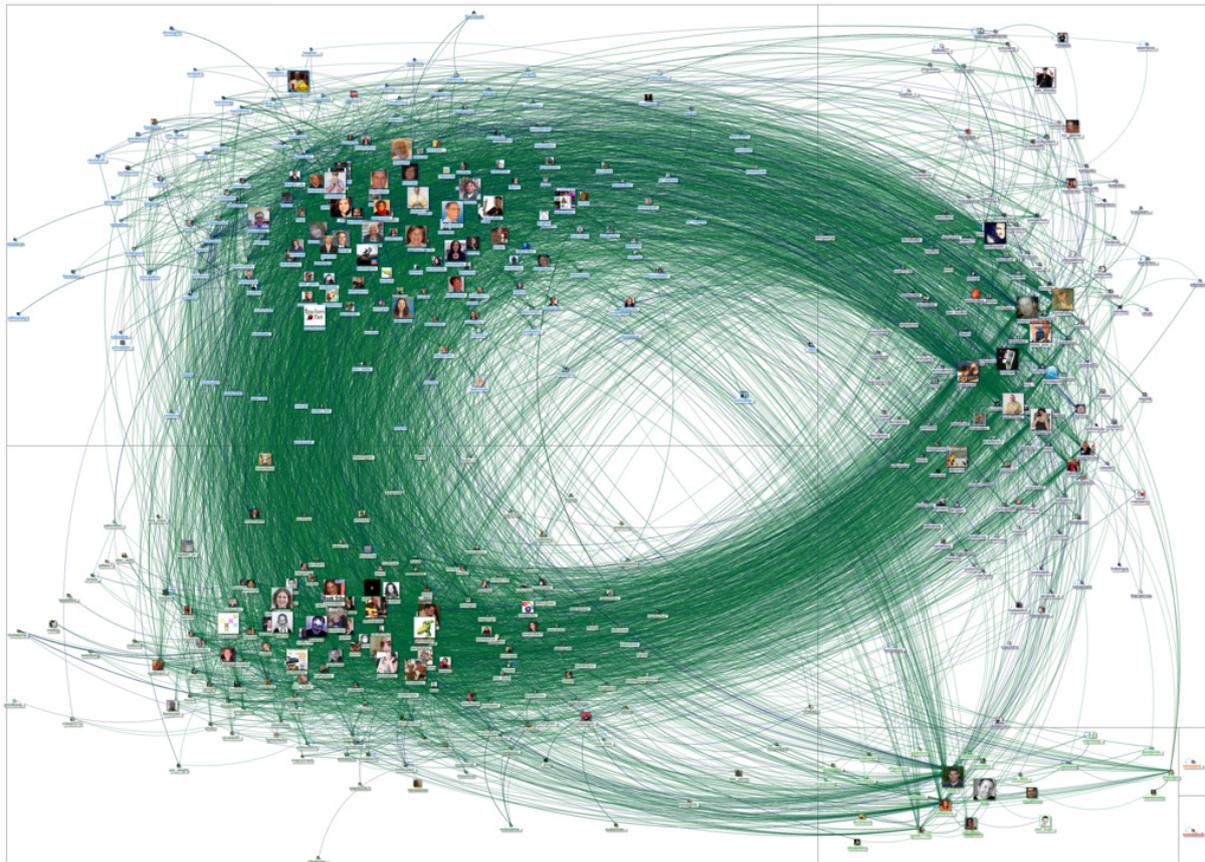


Ada – Sujets des Travaux Dirigés

Algorithmique, second semestre
INSA première année

Social media network connections among Twitter users



Created with NodeXL (<http://nodexl.codeplex.com>) from the Social Media Research Foundation (<http://www.smrffoundation.org>)

Déroulement des séances

- Vous travaillez par équipes de quatre étudiants.

Sur le travail demandé

- Pour résoudre le problème posé, vous pouvez avoir besoin d'utiliser des concepts qui n'ont pas encore été vus en cours. C'est un choix pédagogique : feuillotez le polycopié de cours.
- Pendant la séance, vous pouvez poser des questions de compréhension à l'enseignant, mais il ne résoudra pas le problème à votre place.
- Aucun corrigé officiel ne sera fourni concernant les sujets de TD. Par contre, votre encadrant se fera une joie d'examiner et de commenter ce que vous aurez produit. N'hésitez pas à le solliciter lorsque vous ne comprenez pas un point précis.

Communication numérique et sobriété

Sources :

- Les estimations environnementales proviennent de l'Ademe : impactco2.fr
- Les estimations de débit numérique proviennent de mesures sur Youtube et de negaoctet.org.

Préambule

Sans faire de calcul pour le moment, classez **intuitivement** les activités ci-dessous selon leur impact environnemental dû au trafic de données, ou de manière sensiblement équivalente selon la taille, en octets, des données en transit.

- Un mail avec deux photos envoyé à tous les étudiants INSA Toulouse (3200 étudiants).
- 40 minutes de série Netflix.
- 20 minutes de vidéo 4K sur Youtube.
- 4H de vidéo en basse définition.
- 100 heures de musique ou de podcast audio, sans vidéo.
- 80 000 messages texte (type SMS, ou messagerie Instagram, Whatsapp), longs (avec quelques paragraphes), et comprenant de multiples émojis.

Dans la suite du TD, nous verrons si vous avez vu juste.

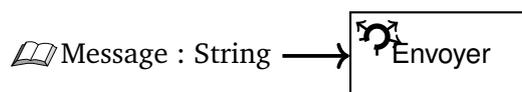
Contexte

Afin de regagner en souveraineté sur le secteur stratégique des réseaux sociaux pour collégiens, votre société a pour mission de développer une nouvelle application de communication rapide et de haut niveau intellectuel : Bikbok.

Votre équipe est chargée d'écrire un démonstrateur capable d'**envoyer** les messages. Une autre équipe se charge de la réception et de l'affichage. Les types des messages sont : texte, vocal, photo, ou vidéo.

1 Message texte

Un bouton "Envoyer" permet l'envoi du message. Ce bouton déclenche la procédure Envoyer ci-dessous, invoquée avec le texte déjà tapé, par exemple "Hello @Laurie, @Paul : pizza ?"



* Écrivez cette procédure en utilisant l'acteur Texte (voir ci-après).

Cette procédure doit compter les destinataires mentionnés dans le message avec '@' (e.g. @Laurie, @Paul), puis envoyer le message. Dégagez l'aide-mémoire du semestre 2, page Tableaux.

Une chaîne de caractères (String) est déjà un tableau :

Indice	1	2	3	4	5	6	7	8	9	10	11	12	...
Message(Indice)	H	e	l	l	o		@	L	a	u	r	i	...

```
package Texte is
```

```
-- Envoie un message. Il faut indiquer le nombre de destinataires.
```

```
procedure Envoyer_texte(Message : String ; Nb_Destinataires : Integer) ;
```

```
end Texte ;
```

- Chaque caractère non accentué occupe un octet. Les autres caractères prennent deux octets. Un émoji occupe deux ou trois octets.
- Un message long de quelques paragraphes contient entre 1000 et 2000 symboles.
- ★ Calculer une estimation haute du nombre d'octets nécessaires pour véhiculer les 80 000 messages du préambule.
- Envoyez-vous 80 000 messages par an?

2 Message vocal

★ Écrire un programme (très court) qui enregistre et envoie un son sans compression, en utilisant l'acteur Vocal ci-dessous.

```
package Vocal is
```

```
-- Un son non compressé est une suite d'échantillons (un tableau d'entiers)
```

```
type T_Tab is array(Integer range <>) of Integer ;
```

```
-- Cette fonction enregistre un message vocal et le renvoie sous forme de tableau T_Tab
```

```
-- Les cases du tableau sont numérotées à partir de 0.
```

```
function Enregistrer_Vocal return T_Tab ;
```

```
-- Envoie un son au destinataire prévu.
```

```
-- Le son peut être compressé ou non compressé.
```

```
procedure Envoyer_Son(Son : T_Tab) ;
```

```
end Vocal ;
```

La musique ou le son se compresse en général très bien sans altérer la qualité. Pour ne pas gaspiller les ressources, cherchons à transmettre moins d'octets.

- Commençons par retirer les silences au début et à la fin du message vocal. L'acteur Compression contient deux fonctions pour les détecter.

- ★ Écrire la fonction Rogner qui renvoie un nouveau tableau, sans les silences de début et de fin.

```
function Rogner(Tab : V.T_Tab) return V.T_Tab is ...
```

- ★ Compresser et envoyer le message vocal rogné.

```
with Vocal ; use Vocal ;
```

```
-- Ce package permet de compresser des morceaux audio dans le format ogg.
```

```
package Compression is
```

```
-- Comprime les données du tableau d'entrée.
```

```
-- Renvoie un nouveau tableau environ 15 fois plus petit.
```

```
function Compresser(Entree : T_Tab) return T_Tab ;
```

```
-- Détection de silence au début du tableau.
```

```
-- Renvoie l'index de fin du silence / début du son.
```

```
function Silence_Debut(Tab : T_Tab) return Integer ;
```

```
-- Idem pour du silence à la fin du tableau
```

```
-- Renvoie l'index de fin du son / début du silence.
```

```
function Silence_Fin(Tab : T_Tab) return Integer ;
```

```
end Compression ;
```

- Le son est échantillonné à 44000 Hz et un échantillon occupe 2 octets (donc 44000×2 octets par seconde avant compression).
- ★ Quel est le rapport de taille entre un message vocal **compressé** et un message texte? (On peut compter 1 minute 30 pour énoncer un texte de 2000 symboles).
- ★ Calculez le volume (en octet) des 100 heures de musique ou de podcast listés dans le préambule (évidemment compressés).
- Quelle durée de messages vocaux envoyez-vous par an?

3 Envoi d'une photo

En utilisant l'acteur Photo, écrire un programme qui :

- ★ Prend une photo,
- ★ Recadre la photo grâce à une fonction que vous écrivez :

```
function Recadrer(Img : T_Image) return T_Image is ...
```



Cette fonction détecte un signe de cadre fait avec deux de vos trois mains et renvoie l'image située à l'intérieur du cadre.

- ★ Comprime l'image, envoie l'image

```
package Photo is
```

```
-- Un pixel coloré.
```

```

type T_Couleur is record
  Rouge, Vert, Bleu : Integer range 0..255 ;
end record ;

-- Une image (une photo) est une matrice
type T_Image is array(Integer range <>, Integer range <>) of T_Couleur ;

-- Prend une photo dès que l'utilisateur presse le bouton.
function Prendre_Photo return T_Image ;

-- Coordonnées d'un cadre (deux coins opposés).
type T_Cadre is record
  -- Position du premier coin
  X1, Y1 : Integer ;
  -- Position du coin opposé, avec X1 < X2 et Y1 < Y2
  X2, Y2 : Integer ;
end record ;

-- Renvoie les coordonnées du cadre détecté dans l'image
-- (sinon renvoie les coins de l'image entière).
function Cadre(Img : T_Image) return T_Cadre ;

function Compresser(Img : T_Image) return T_Image ;

-- Envoie une image compressée ou non.
procedure Envoyer_Image(Img : T_Image) ;

end Photo ;

```

La photo est une matrice de taille 1920x1080. Chaque cellule contient un pixel coloré et occupe 3 octets (rouge, vert, bleu). Le taux de compression d'une image peut atteindre 15 sans perdre trop qualité.

- ★ Quel est le rapport de taille entre une photo compressée et un message texte (de 2000 symboles) ?
- ★ Hors sujet : quel est le rapport de taille entre la photo d'un écran avec un programme Ada et le même programme Ada envoyé en pièce jointe ? (par exemple un programme de 2800 octets)
- ★ Estimez le trafic total engendré par un mail avec deux photos envoyé aux 3200 étudiants de l'INSA (même si au final, seulement 40 étudiants seront concernés).
- Combien de photos transmettez-vous par an ? (noter que chaque photo prise est probablement envoyée sur votre drive)

4 Message vidéo

Pour terminer de comparer les éléments du préambule, voici les ordres de grandeur des débits vidéos :

- Série Netflix : 1 000 ko/s (kilo-octet)
- Youtube 4K : 2 000 ko/s
- Youtube basse définition : 150 ko/s

n = 1 Mo / s

★ Maintenant que vous avez toutes les données, faites le classement exact.

5 Responsabilité de l'équipe de développement

- ✗ Nous visons à terme un million d'utilisateurs. Calculer le trafic de données engendré à chaque fois que nous mettrons à jour l'application – une mise à jour nécessite que chaque utilisateur télécharge l'application, soit 45Mo (méga-octets). Conclusion ?
- ✗ Quelle version de système d'exploitation (e.g. Android) utiliser ? Toujours la version à jour du moment, ou une version plus ancienne ? (Quel est l'enjeu derrière ce choix ?)

6 Responsabilité personnelle vs émission de gaz à effet de serre

Le rôle de chacun dans l'émission de gaz à effet de serre -et de contribution au dérèglement climatique- peut se mesurer en masse équivalent CO₂ (kg CO₂e).

Faites un petit calcul :

- Idéalement, chacun devrait tendre vers 1T de CO₂e par an.¹. Cette tonne CO₂e couvre l'alimentation, le transport, l'habillement, le chauffage, etc. et l'usage numérique.
- Parmi ce quota de 1T, fixons de manière arbitraire un quota personnel numérique à 3%, comprenant uniquement l'usage numérique (trafic de données). La partie achat de matériel (ordi, tablette, smartphone, enceinte bluetooth, casque audio, ...) doit être comptée dans un autre poste. Une console de jeux, par exemple, compte pour environ 100kg CO₂e.
- Ces 3% de données numériques représentent donc 30kg CO₂e par an. q = 30 kg
- On essaie de répartir ces 30kg CO₂e équitablement sur l'année.

Video	Pause	Video	Pause	Video	Pause...
18g		6g		24g	

- Vous regardez une série Netflix de 40 minutes. Calculer le coût CO₂e, sachant que le trafic de données représente 10g CO₂e/Go (10 grammes par giga-octet). d = 40 min ρ = 10g / Go
Ne tombez pas dans le piège grossier de M. Le Botlan des unités incompatibles (g/kg, Mo/Go, s/min)
- L'idée est de faire une pause pour laisser votre moyenne de coût CO₂e numérique redescendre : pendant la pause, pas d'utilisation significative du réseau, pas de video, pas de Tiktok.
- ★ Combien de temps doit durer la pause avant un autre usage numérique (une autre video par exemple) pour garantir que vous ne dépasserez pas 30kg CO₂e sur l'année ?
- ★ Faites le même calcul avec une video de qualité réduite, le résultat est intéressant !

Le même genre de raisonnement peut s'appliquer à d'autres activités, dès lors que vous avez décidé de votre quota personnel sur l'année.

1. On parle habituellement de 2T, mais la moitié concerne la partie "infrastructure", i.e. tout ce qui est construit, entretenu, utilisé par l'État, et finalement collectivement utile à l'ensemble des habitants.

TD – Problème 2 : structures de données imbriquées

Contexte : Plongée

La FFESSM (Fédération Française d'Études et de Sports Sous-Marins) nous sollicite pour mettre en place un outil internet permettant d'éditer des *fiches de sécurité* de plongée. La fiche de sécurité est un document obligatoire pour toute sortie en mer d'un groupe de plongeurs et est régulièrement contrôlée sur site par la gendarmerie maritime.

L'acteur à utiliser est fourni (cf. page 10). Un exemple de fiche de sécurité se trouve en quatrième de couverture.

Ada n'étant pas pratique pour manipuler des chaînes de caractères, les noms des plongeurs sont réduits à des **trigrammes** (identifiant de trois lettres), par exemple LUK ou HAN.



Vocabulaire

Pour les besoins du TD, les notions sont volontairement simplifiées (l'acteur ne contient pas toutes les informations visibles sur la fiche d'exemple).

- **Palanquée** : Petite équipe de plongeurs qui restent ensemble pendant la plongée.
- **Rôle** : Rôle du plongeur au sein de sa palanquée (Enseignant, Élève, Guide de Palanquée, plongeur en exploration Encadrée, plongeur Autonome)
- **Type de plongée** (visible sur la fiche, mais non défini par l'acteur) : plongée Technique (enseignement), plongée Autonome, plongée Encadrée (exploration accompagnée)



Travail demandé

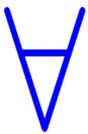
Le formulaire internet doit afficher automatiquement le nombre de plongeurs inscrits, et signaler si les rôles attribués sont incohérents entre eux. Pour cela vous devez définir plusieurs fonctions :

- Écrire une fonction `Effectif_Palanquee` qui reçoit en argument une palanquée et compte le nombre de plongeurs de cette palanquée, puis écrire une fonction `Effectif_Fiche` qui reçoit une fiche et compte le nombre total de plongeurs présents dans la fiche.
- Écrire une fonction `Est_Plongee_Technique` qui vérifie qu'une palanquée correspond bien à une plongée Technique : la fonction reçoit en argument une palanquée P et évalue l'assertion « P contient exactement : un enseignant et 1 à 4 élèves».

Algorithme de recherche



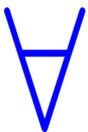
- Écrivez une fonction `Numero_Premiere_Technique` qui prend en argument une fiche de sécurité et qui renvoie le numéro de la première palanquée de la liste de type **technique** (en utilisant `Est_Plongee_Technique`). La fonction renvoie 0 si aucune palanquée ne convient.
 - Votre fonction doit s'arrêter de parcourir la fiche dès que la palanquée est trouvée.
 - La fonction ne contient qu'un seul **return**, situé à la fin.



- Écrivez une fonction `Durees_Fiche_OK` qui vérifie que la durée de plongée de chaque palanquée ne dépasse pas la durée de plongée maximale autorisée par la fiche de sécurité. La fonction renvoie un booléen.
(Il se trouve que c'est aussi un algorithme de recherche.)



- Écrivez une fonction `Profondeur_Max_Plongeurs` qui reçoit en argument un tableau de plongeurs et calcule la profondeur maximale à laquelle ces plongeurs peuvent aller. *Question à deux centimes : est-ce un algorithme de min ou de max?*



- Écrivez une fonction `Profondeurs_OK` qui prend en argument une fiche de sécurité et qui vérifie que tous les plongeurs de toutes les palanquées ont bien respecté leur profondeur maximale autorisée. La fonction renvoie un booléen. (Tâchez d'impressionner votre encadrant en utilisant `Profondeur_Max_Plongeurs`).

```
--
-- Acteur représentant certaines informations des fiches de sécurité en plongée.
--
package Fiche_Plongee is

  -- Trigramme pour nommer chaque plongeur
  subtype T_Trigramme is String(1..3) ;

  -- Role du plongeur dans sa palanquée.
  type T_Role is (Ens, Eleve, GP, Enca, Auto) ;

  type T_Plongeur is record
    -- Ce booléen indique si la case du tableau est utilisée.
    -- S'il est faux, les autres attributs n'ont pas de signification
    -- car ou bien le plongeur est absent, ou bien la ligne de la fiche est vide.
    Present: Boolean ;

    -- Informations utiles lorsque le plongeur est présent.
    Nom: T_Trigramme ;
    Profondeur_Max_Autorisee: Integer ;
    Role: T_Role ;
  end record ;

  -- Ensemble de plongeurs
  type T_Tab_Plongeurs is array(Integer range <>) of T_Plongeur ;

  -- Nombre maximal de plongeurs dans une palanquée
  Max_Plongeurs: Integer := 5 ;

  type T_Palanq is record
    -- Durée passée au fond, en minutes
    Duree_Reelle_Fond: Integer ;

    -- Profondeur maximale de plongée, en mètres
    Profondeur_Reelle: Integer ;

    -- Tableau des plongeurs
    Plongeurs: T_Tab_Plongeurs(1..Max_Plongeurs) ;
  end record ;

  type T_Tab_Palanquees is array(Integer range <>) of T_Palanq ;

  -- Nombre maximal de palanquées inscrites sur la fiche
  Max_Palanquees: Integer := 10 ;

  type T_Fiche_Secu is record

    -- Directeur de plongée
    Directeur: T_Trigramme ;

    -- Tableau contenant les palanquées.
    -- Toutes les cases ne sont pas utilisées.
    Tab_Palanquees: T_Tab_Palanquees(1..Max_Palanquees) ;

    -- Nombre exact de palanquées utilisées dans le tableau.
    -- Les cases utilisées sont les cases de 1 à Nb_Palanquees.
    NB_Palanquees: Integer ;

    -- Temps maximal autorisé par le directeur de plongée, en minutes
    Temps_Plongee_Max_Autorise: Integer ;
  end record ;
```

TD – Problème 3 : gestion d'événements asynchrones

26 juillet 2024, Paris

Ce matin a eu lieu la première épreuve de natation des jeux olympiques, à l'Aréna la Défense. À cause d'un dysfonctionnement logiciel, le chronométrage officiel s'est arrêté en pleine course, au moment où le nageur handisport Théo Curin, alors en tête, a effectué son premier demi-tour. De plus, il a fallu attendre 25 minutes pour réussir à départager trois nageurs arrivés en même temps : les juges ont dû visionner sur Tik Tok une vidéo de la course réalisée par un spectateur (malheureusement en selfie).



Dès midi, les organisateurs des J.O. ont lancé un appel d'offres pour réécrire intégralement le logiciel de chronométrage des épreuves de natation. Le délai est de 18 heures, car il faut être prêt pour l'épreuve du 400 mètres nage libre ayant lieu demain.

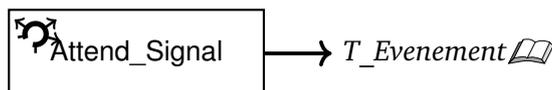
En vacances avec quelques collègues de l'INSA, vous décidez de répondre à cet appel d'offres pour occuper constructivement votre début d'après-midi et encaisser au passage les 32 000 €.

Cahier des charges

Objectifs Écrire un programme qui compte les aller-retours de chaque nageur jusqu'à 400 mètres, et affiche les numéros de couloir des nageurs dans l'ordre d'arrivée, avec leur numéro d'arrivée. Le programme est aussi capable de chronométrer chaque nageur.

Une piscine olympique fait 50 mètres de long. Elle peut avoir un nombre quelconque de couloirs utilisés (en général 8). Pour spécifier le nombre de couloirs, il suffira de modifier une seule ligne de votre programme.

Capteurs Les capteurs envoient des signaux qui sont immédiatement enregistrés puis peuvent être récupérés par la fonction bloquante `Attend_Signal` (voir dans l'acteur Capteurs ci-dessous).



- Lorsque le nageur de la ligne numéro i quitte son plot de départ, le capteur situé dans le plot envoie un signal de type `T_Evenement`, dans lequel `Emetteur = Capteur_Plot`, `Num = i`, et `Date =` la date de l'événement en secondes.
- De même, lorsque le nageur de la ligne numéro i touche l'un des bords du bassin, les capteurs optiques envoient un signal. Les plots de départ sont sur le côté Nord. Le capteur étant très sensible, plusieurs signaux peuvent être envoyés d'affilée pour un seul toucher d'un nageur.
- Le signal de départ est signalé par l'événement `Signal_Feu`.
- Enfin, l'arbitre indique la fin de la course avec un bouton qui envoie le signal `Bouton_Fin`.

Démarche

- Écrivez sur papier une suite d'événements possibles pour une course de 100m à trois nageurs. Notez les événements comme ceci : `(Feu, 32400.0)` `(Plot, 4, 32400.3)` `(Sud, 2, 32423.6)`, etc.
- Dessinez un algorithme de votre programme sur papier.

Pour améliorer

- Détecter les faux départs.
- Pour améliorer l’affichage, votre programme peut demander le nom des nageurs au début. Un type **record** ne peut contenir que des chaînes (*String*) de taille fixée, ce qui s’écrit `String (1..12)` pour une chaîne de 12 caractères.

Vous pourrez utiliser la fonction `Normalise (X, N)` de `GAda.Text_IO`, qui prend une chaîne `X` et renvoie une chaîne de taille `N` en la tronquant ou en la complétant avec des espaces.

- Pour chaque nageur, repérer son meilleur temps de parcours de 50m (sur une longueur de piscine). À la fin de l’épreuve, lorsque tous sont arrivés, afficher un classement des nageurs les plus rapides sur 50m.

`capteurs.ads`

```
--
-- Cet acteur permet de recevoir des signaux provenant des capteurs situés autour du bassin.
--
package Capteurs is

  -- Énumération des différentes sortes de capteurs
  -- | Capteur_Plot : capteur situé sur le plot de départ
  -- | Capteur_Nord : capteur situé à l’extrémité Nord du bassin
  -- | Capteur_Sud : capteur situé à l’extrémité Sud du bassin
  -- | Signal_Feu : signal émis au moment du départ
  -- | Bouton_Fin : bouton de fin de course, activé par l’arbitre
  type T_Capteur is (Capteur_Plot, Capteur_Nord, Capteur_Sud, Signal_Feu, Bouton_Fin) ;

  -- Un événement est un signal émis par un capteur
  -- Numero est le numéro de couloir du capteur (pour les capteurs Plot, Nord, Sud)
  -- Date est la date de l’événement, en secondes depuis minuit.
  type T_Evenement is record
    Emetteur : T_Capteur ;
    Num : Integer ;
    Date : Float ;
  end record ;

  -- Attend en bloquant le programme jusqu’à ce qu’un événement se déclenche,
  -- et renvoie l’événement qui s’est produit.
  -- Les événements arrivent forcément dans l’ordre chronologique.
  function Attend_Signal return T_Evenement ;

end Capteurs ;
```

Version du 14 janvier 2025.

FICHE DE SECURITE (Plongée du bord)						SECOURS : Tél. 15 (SAMU), 112 (EU) ; VHF canal 16			
Date :		22/10/2016 (Après-midi)		Site, Lieu :		Le Cimetière, Port-Vendres			
DP :		Dominique LONGIN		Sécurité		1er tour :			
		6 palanquées / 17 plongeurs		Surface		2e tour			
Consignes : <input type="checkbox"/> Respect des prérog. <input type="checkbox"/> Retour surface avec 50 bar min. <input type="checkbox"/> Parachute obligatoire <input type="checkbox"/> Perte palanquée									
Matériel sécu : <input type="checkbox"/> 1 parachute/pal. min. <input type="checkbox"/> GP = 2 dét. complets <input type="checkbox"/> Auton. OU prof. > 20 m => octopus <input type="checkbox"/> Moyens de déco									
N° pal.	Type	Nom	Prénom	Aptitude	Fonct° (*)	Commentaires	Paramètres réels		
1 (1er tour)	Tech	L	Dominique	E3	E		Durée fond	Prof. Max.	Paliers
		F	Adriana	PE20	El		Prévue : 50'	Prévue : 20m	
		C	Bertrand	PE20	El		Réelle : 36'	Réelle : 16m	
							Heure départ	14:36	
					Heure retour	15:02			
2 (1er tour)	Explo. auton.	T	Christophe	PA60	A		Durée fond	Prof. Max.	Paliers
		M	Jerome	PA60	A		Prévue : 50'	Prévue : 40m	
							Réelle : 41'	Réelle : 21m	
							Heure départ	14:13	
					Heure retour	15:46			
3 (1er tour)	Tech	H	Alex	E3	E		Durée fond	Prof. Max.	Paliers
		F	Cyndia	PE40	El		Prévue : 50'	Prévue : 40m	
		B	Sébastien	PE40	El		Réelle : 48'	Réelle : 26m	
							Heure départ	14:26	
					Heure retour	15:17			
4 (1er tour)	Tech	Cl	Marc	E3	E		Durée fond	Prof. Max.	Paliers
		O	Julien	PE40	El		Prévue : 50'	Prévue : 40m	
		C	William	PE40	El		Réelle : 36'	Réelle : 23m	
							Heure départ	14:22	
					Heure retour	14:59			
5 (1er tour)	Explo. auton.	D	Thomas	PA20	A		Durée fond	Prof. Max.	Paliers
		H	François	PA20	A		Prévue : 50'	Prévue : 20m	
		B	Bertrand	PA20	A		Réelle : 42'	Réelle : 20m	
							Heure départ	14:13	
					Heure retour	14:57			
6 (1er tour)	Explo. encad.	L	Sébastien	E4	GP		Durée fond	Prof. Max.	Paliers
		S	Audrey	PE40	Enc		Prévue : 50'	Prévue : 20m	
		B	Marjorie	PE40	Enc		Réelle :	Réelle :	
							Heure départ		
					Heure retour				
7							Durée fond	Prof. Max.	Paliers
							Prévue : 50'	Prévue : 20m	
							Réelle :	Réelle :	
							Heure départ		
					Heure retour				
8							Durée fond	Prof. Max.	Paliers
							Prévue : 50'	Prévue : 20m	
							Réelle :	Réelle :	
							Heure départ		
					Heure retour				
9							Durée fond	Prof. Max.	Paliers
							Prévue : 50'	Prévue : 20m	
							Réelle :	Réelle :	
							Heure départ		
					Heure retour				
10							Durée fond	Prof. Max.	Paliers
							Prévue : 50'	Prévue : 20m	
							Réelle :	Réelle :	
							Heure départ		
					Heure retour				

(*) E (Enseignant) ; EL (élève) ; GP (guide de palanquée) ; Enc (encadré en explo) ; A (autonome) ; SF (serre-file)