



LESSONSHARING SAS

Exercices Algorithmie Correction

Novembre 2020



www.lessonsharing.fr

Exercice 1

Écrire un algorithme qui lira au clavier l'heure et les minutes, et affichera l'heure qu'il sera une minute plus tard.

Exemple :

Si l'utilisateur tape 15 puis 30, l'algorithme doit répondre : "Dans une minute, il sera 15h31".

Note : on suppose que l'utilisateur entre une heure valide. Pas de vérification nécessaire.

Vous utiliserez des fonctions natives appelées **AFFICHER** (pour simuler l'affichage un message) et **LIRE** (pour simuler une entrée d'un utilisateur au clavier).

ALGORITHME *prochaine_minute*

VARIABLES

entier : h, m

DEBUT

AFFICHER ("Entrez les heures, puis les minutes : ")

LIRE (h, m)

$m \leftarrow m + 1$

SI $m = 60$ **ALORS**

$m \leftarrow 0$

$h \leftarrow h + 1$

FIN SI

Si $h = 24$ **ALORS**

$h \leftarrow 0$

FIN SI

AFFICHER ("Dans une minute il sera ", h, "heure(s) ", m, "minute(s)")

FIN

Exercice 2

De même que le précédent, cet algorithme doit demander une heure et en afficher une autre. Mais cette fois, il doit gérer également les secondes, et afficher l'heure qu'il sera une seconde plus tard.

Exemple :

Si l'utilisateur tape 15, puis 30, puis 10, l'algorithme doit répondre : "Dans une seconde, il sera 15h30 et 11 sec".

Note : là encore, on suppose que l'utilisateur entre une date valide.

ALGORITHME *prochaine_seconde*

VARIABLES

entier : h, m, s

DEBUT

AFFICHER ("Entrez les heures, puis les minutes, puis les secondes : ")

LIRE (h, m, s)

$s \leftarrow s + 1$

SI $s = 60$ **ALORS**

$s \leftarrow 0$

$m \leftarrow m + 1$

FIN SI

SI $m = 60$ **ALORS**

$m \leftarrow 0$

$h \leftarrow h + 1$

FIN SI

Si $h = 24$ **ALORS**

$h \leftarrow 0$

FIN SI

AFFICHER ("Dans une minute il sera ", h, "heure(s) ", m, " minute(s) et ", s, " secondes")

FIN

Exercice 3

Un magasin d'impressions facture 0,10 E les dix premières photocopies, 0,09 E les vingt suivantes et 0,08 E au-delà. Ecrivez un algorithme qui demande à l'utilisateur le nombre de photocopies effectuées et qui affiche la facture correspondante.

ALGORITHME *facture_photocopie*

VARIABLES

entier : n, p

DEBUT

AFFICHER ("Nombre de photocopies : ")

LIRE (n)

SI $n \leq 10$ **ALORS**

$p \leftarrow n * 0,1$

SINON SI $n \leq 30$ **ALORS**

$p \leftarrow 10 * 0,1 + (n - 10) * 0,09$

SINON

$p \leftarrow 10 * 0,1 + 20 * 0,09 + (n - 30) * 0,08$

FIN SI

AFFICHER ("Le prix total est: ", p)

FIN

Exercice 4

Les habitants d'un pays atypique paient l'impôt selon les règles suivantes :

- Les hommes de plus de 20 ans paient l'impôt
- Les femmes paient l'impôt si elles ont entre 18 et 35 ans
- Les autres ne paient pas d'impôt

Le programme demandera donc l'âge et le sexe de l'utilisateur, et se prononcera donc ensuite sur le fait que l'habitant est imposable.

ALGORITHME *impots*

VARIABLES

chaîne : sexe
entier : age
boolean : cas1, cas2

DEBUT

AFFICHER ("Entrez le sexe (M/F) : ")
LIRE (*sexe*)
AFFICHER ("Entrez l'âge: ")
LIRE (*age*)
cas1 ← *sexe* = "M" ET *age* > 20
cas2 ← *sexe* = "F" ET (*age* > 18 ET *age* < 35)
SI *cas1* **OU** *cas2* **ALORS**
 AFFICHER ("Imposable")
Sinon
 AFFICHER ("Non Imposable")
FIN SI

FIN

Exercice 5

Les élections législatives d'un pays atypique, obéissent à la règle suivante :

- Lorsque l'un des candidats obtient plus de 50% des suffrages, il est élu dès le premier tour.
- En cas de deuxième tour, peuvent participer uniquement les candidats ayant obtenu au moins 12,5% des voix au premier tour.

Vous devez écrire un algorithme qui permette la saisie des scores de quatre candidats au premier tour. Cet algorithme traitera ensuite le candidat numéro 1 (et **uniquement** lui) : il dira s'il est élu, battu, s'il se trouve en ballottage favorable (il participe au second tour en étant arrivé en tête à l'issue du premier tour) ou défavorable (il participe au second tour sans avoir été en tête au premier tour).

ALGORITHME *elections_legislatives*

VARIABLES

entier: p1, p2, p3, p4

boolean: cas1, cas2, cas3, cas4

DEBUT

AFFICHER ("Entrez les scores des quatre prétendants :")

LIRE (p1, p2, p3, p4)

cas1 \leftarrow p1 > 50

cas2 \leftarrow p2 > 50 ou p3 > 50 ou p4 > 50

cas3 \leftarrow p1 \geq p2 et p1 \geq p3 et p1 \geq p4

cas4 \leftarrow p1 \geq 12,5

SI cas1 **ALORS**

AFFICHER ("Elu au premier tour")

SINON SI cas2 **OU NON** cas4 **ALORS**

AFFICHER ("Battu, éliminé, sorti !!!")

SINON SI cas3 **ALORS**

AFFICHER ("Ballottage favorable")

SINON

AFFICHER ("Ballottage défavorable")

FIN SI

FIN

Exercice 6

Une compagnie d'assurance automobile propose à ses clients quatre familles de tarifs identifiables par une couleur, du moins au plus onéreux : tarifs bleu, vert, orange et rouge. Le tarif dépend de la situation du conducteur :

- Un conducteur de moins de 25 ans et titulaire du permis depuis moins de deux ans, se voit attribuer le tarif rouge, si toutefois il n'a jamais été responsable d'accident. Sinon, la compagnie refuse de l'assurer.
- Un conducteur de moins de 25 ans et titulaire du permis depuis plus de deux ans, ou de plus de 25 ans mais titulaire du permis depuis moins de deux ans a le droit au tarif orange s'il n'a jamais provoqué d'accident, au tarif rouge pour un accident, sinon il est refusé.
- Un conducteur de plus de 25 ans titulaire du permis depuis plus de deux ans bénéficie du tarif vert s'il n'est à l'origine d'aucun accident et du tarif orange pour un accident, du tarif rouge pour deux accidents, et refusé au-delà
- De plus, pour encourager la fidélité des clients acceptés, la compagnie propose un contrat de la couleur immédiatement la plus avantageuse s'il est entré dans la maison depuis plus de cinq ans. Ainsi, s'il satisfait à cette exigence, un client normalement "vert" devient "bleu", un client normalement "orange" devient "vert", et le "rouge" devient orange.

Ecrire l'algorithme permettant de saisir les données nécessaires (sans contrôle de saisie) et de traiter ce problème.

ALGORITHME *situation_assurance*

VARIABLES

entier: age, permis, accidents, nb_annees_assurance, penalite = 0

booleen: cas1, cas2, cas3

chaîne: situation

DEBUT

AFFICHER ("Entrez l'âge: ")

LIRE (age)

AFFICHER ("Entrez le nombre d'années de permis: ")

LIRE (permis)

AFFICHER ("Entrez le nombre d'accidents: ")

LIRE (accidents)

AFFICHER ("Entrez le nombre d'années d'assurance: ")

LIRE (nb_annees_assurance)

$\text{cas1} \leftarrow \text{age} \geq 25$

$\text{cas2} \leftarrow \text{permis} \geq 2$

$\text{cas3} \leftarrow \text{nb_annees_assurance} > 5$

SI NON cas1 **ALORS**

$\text{penalite} \leftarrow \text{penalite} + 1$

FIN SI

SI NON cas2 **ALORS**

$\text{penalite} \leftarrow \text{penalite} + 1$

FIN SI

$\text{penalite} \leftarrow \text{penalite} + \text{accidents}$

SI $\text{penalite} < 3$ **ET** cas3 **ALORS**

$\text{penalite} \leftarrow \text{penalite} - 1$

FIN SI

SELON penalite

$\text{penalite} = -1$: situation \leftarrow "Bleu"

$\text{penalite} = 0$: situation \leftarrow "Vert"

$\text{penalite} = 1$: situation \leftarrow "Orange"

$\text{penalite} = 2$: situation \leftarrow "Rouge"

AUTREMENT

 situation \leftarrow "Refusé"

FIN SELON

AFFICHER ("Votre situation : ", situation)

FIN



Exercice 7

Ecrire un algorithme qui demande à l'utilisateur un nombre compris entre 1 et 10 jusqu'à ce que la réponse convienne.

ALGORITHME *saisie_entre_un_et_dix*

VARIABLES

entier : n

DEBUT

$n \leftarrow 0$

ECRIRE ("Entrez un nombre entre 1 et 10")

REPETER

LIRE (n)

SI $n < 1$ **OU** $n > 10$

AFFICHER ("Saisie erronée. Recommencez")

FIN SI

JUSQU A $n > 1$ **OU** $n < 10$

FIN

Exercice 8

Ecrire un algorithme qui demande un nombre compris entre 10 et 20, jusqu'à ce que la réponse convienne. En cas de réponse supérieure à 20, on fera apparaître un message: « Plus petit ! », et inversement, « Plus grand ! » si le nombre est inférieur à 10.

ALGORITHME *saisie_entre_un_et_dix*

VARIABLES

entier : n

DEBUT

$n \leftarrow 0$

ECRIRE ("Entrez un nombre entre 1 et 10")

TANT QUE $n < 10$ **OU** $n > 20$

LIRE (n)

SI $n < 10$ **ALORS**

AFFICHER ("Plus grand !")

Si $n > 20$ **ALORS**

AFFICHER ("Plus petit !")

FIN SI

FIN TANT QUE

FIN

Exercice 9

Ecrire un algorithme qui demande un nombre de départ, et qui ensuite affiche les dix nombres suivants. Par exemple, si l'utilisateur entre le nombre 17, le programme affichera les nombres de 18 à 27.

ALGORITHME *dix_nombres_suivants*

VARIABLES

entier : n, i = 0

DEBUT

AFFICHER ("Entrez un nombre : ")

LIRE (n)

AFFICHER ("Les 10 nombres suivants sont : ")

POUR i **ALLANT DE** 0 **à** 10 **PAR PAS DE** 1

AFFICHER (n + i)

FIN POUR

FIN

Exercice 10

Ecrire un algorithme qui demande un nombre de départ, et qui ensuite écrit la table de multiplication de ce nombre, présentée comme suit (cas où l'utilisateur entre le nombre 8) :

8 x 1 = 8

8 x 2 = 16

...

8 x 10 = 80

ALGORITHME *afficher_table*

VARIABLES

entier : n, i

DEBUT

AFFICHER ("Entrez un nombre : ")

LIRE (n)

AFFICHER ("La table de multiplication de ce nombre est : ")

POUR i **ALLANT DE** 1 **à** 10 **PAR PAS DE** 1

AFFICHER (n, "x", i, "=", n * i)

FIN POUR

FIN

Exercice 11

Ecrire un algorithme qui demande un nombre de départ, et qui calcule la somme des entiers jusqu'à ce nombre. Par exemple, si l'on entre 5, le programme doit calculer : $1 + 2 + 3 + 4 + 5 = 15$

Note: on souhaite afficher uniquement le résultat, pas la décomposition du calcul.

ALGORITHME *somme_des_entiers_dans_n*

VARIABLES

entier : n, i, somme = 0

DEBUT

AFFICHER ("Entrez un nombre : ")

LIRE (n)

POUR i **ALLANT DE** 1 à n **PAR PAS DE** 1

somme \leftarrow somme + i

FIN POUR

AFFICHER ("La somme des entiers qui composent ", n, " est : ", somme)

FIN

Exercice 12

Ecrire un algorithme qui demande un nombre de départ, et qui calcule sa factorielle.

NB : la factorielle de 8, notée 8 ! vaut

$1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 8$

ALGORITHME *factorielle_de_n*

VARIABLES

entier : n, i, factorielle = 1

DEBUT

AFFICHER ("Entrez un nombre : ")

LIRE (n)

POUR i **ALLANT DE** 2 à n **PAR PAS DE** 1

Factorielle \leftarrow factorielle * i

FIN POUR

AFFICHER ("La factorielle est : ", factorielle)

FIN

Exercice 13

Ecrire un algorithme qui demande successivement 20 nombres à l'utilisateur, et qui lui dise ensuite quel était le plus grand parmi ces 20 nombres, ainsi que sa position :

Ex :

Entrez le nombre numéro 1 : 12

Entrez le nombre numéro 2 : 14

etc.

Entrez le nombre numéro 20 : 6

Le plus grand de ces nombres est : 14 à la position numéro 2

ALGORITHME *plus_grand_parmis_20_nombres*

VARIABLES

entier : n, i, plus_grand_n = 0, position_plus_grand_n

DEBUT

POUR i ALLANT DE 1 à 20 PAR PAS DE 1

AFFICHER ("Entrez un nombre : ")

LIRE (n)

SI i = 1 **OU** n > plus_grand_n **ALORS**

plus_grand_n ← n

position_plus_grand_n ← i

FIN SI

FIN POUR

AFFICHER ("Le plus grand est : ", plus_grand_n, "à la position ", position_plus_grand_n)

FIN

Exercice 14

Réécrire l'algorithme précédent, mais cette fois-ci on ne connaît pas d'avance combien l'utilisateur souhaite saisir de nombres. La saisie des nombres s'arrête lorsque l'utilisateur entre un zéro.

ALGORITHME *plus_grand_parmis_n_nombres*

VARIABLES

entier : n, i = 0, plus_grand_n = 1, position_plus_grand_n

DEBUT

TANT QUE n <> 0

AFFICHER ("Entrez un nombre : ")

LIRE (n)

i ← i + 1

SI i = 1 **OU** n > plus_grand_n **ALORS**

plus_grand_n ← n

position_plus_grand_n ← i

FIN SI

FIN TANT QUE

AFFICHER ("Le plus grand est : ", plus_grand_n, "à la position ", position_plus_grand_n)

FIN

Exercice 15

Lire la suite des prix (en euros entiers et terminée lorsque l'utilisateur entre un zéro) des achats d'un client. Calculer la somme qu'il doit, lire la somme qu'il paye, et simuler la remise de la monnaie en affichant les textes "10 Euros", "5 Euros" et "1 Euro" autant de fois qu'il y a de coupures de chaque sorte à rendre.

ALGORITHME *calculer_monnaie*

VARIABLES

entier : euros = 0, somme_due = 0, montant_verse = 0, reste
entier : nb_10_eu, nb_5_eu, nb_1_eu

DEBUT

REPETER

AFFICHER ("Entrez un montant : ")

LIRE (euros)

$\text{somme_due} \leftarrow \text{somme_due} + \text{euros}$

JUSQU A $e = 0$

AFFICHER ("somme due en euros : ", somme_due)

LIRE (montant_verse)

$\text{reste} \leftarrow \text{montant_verse} - \text{somme_due}$

$\text{nb_10_eu} \leftarrow 0$

TANT QUE $\text{reste} \geq 10$

$\text{nb_10_eu} \leftarrow \text{nb_10_eu} + 1$

$\text{reste} \leftarrow \text{reste} - 10$

FIN TANT QUE

$\text{nb_5_eu} \leftarrow 0$

TANT QUE $\text{reste} \geq 5$

$\text{nb_5_eu} \leftarrow \text{nb_5_eu} + 1$

$\text{reste} \leftarrow \text{reste} - 5$

FIN TANT QUE

AFFICHER ("Rendu de la monnaie :")

AFFICHER ("Billets de 10 euros : ", nb_10_eu)

AFFICHER ("Billets de 5 euros : ", nb_5_eu)

AFFICHER ("Pièces de 1 euros : ", reste)

FIN