2) Le tour de magie : "L'âge automatique" :

Depuis : http://www.juggling.ch/gisin/program/blockly/abgex/ex2000/ex2000.html
Configuration > Divers exemples, au choix... > Choix 3.

Programmez l'algorithme suivant :

a) Pensez au nombre de soirs que vous voulez sortir par semaine, en moyenne (au cinéma ou théâtre ou au resto ou voir un ami ou autre...).
b) Placez ce nombre une variable, (z par exemple).

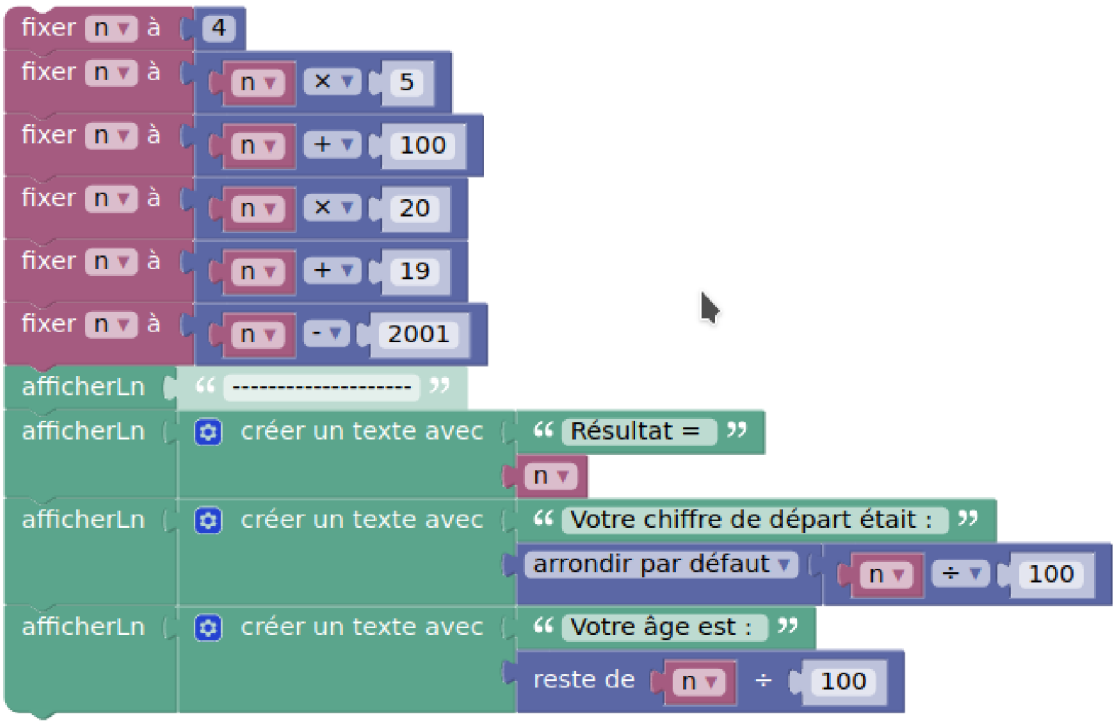
Vous avez ainsi un chiffre en tête.
° Multipliez par 5 votre chiffre
° Additionnez 100
° Multipliez par 20
° Additionnez 19
° Si vous avez déjà eu votre anniversaire cette année, additionnez 1.
° Soustrayez votre année de naissance.

Vous obtenez tous un nombre de 3 chiffres !
° Le chiffre des centaines est le nombre de fois que vous désirez sortir le soir par semaine !
° Les deux autres chiffres indiquent votre âge !

c) Saurez-vous montrer mathématiquement les deux affirmations précédentes ?

d) Affichez ces deux constatations.
° Saurez-vous extraire du nombre de 3 chiffres, le chiffre des centaines ?
° Saurez-vous extraire du nombre de 3 chiffres, les deux premiers chiffres ?

Corrigé :



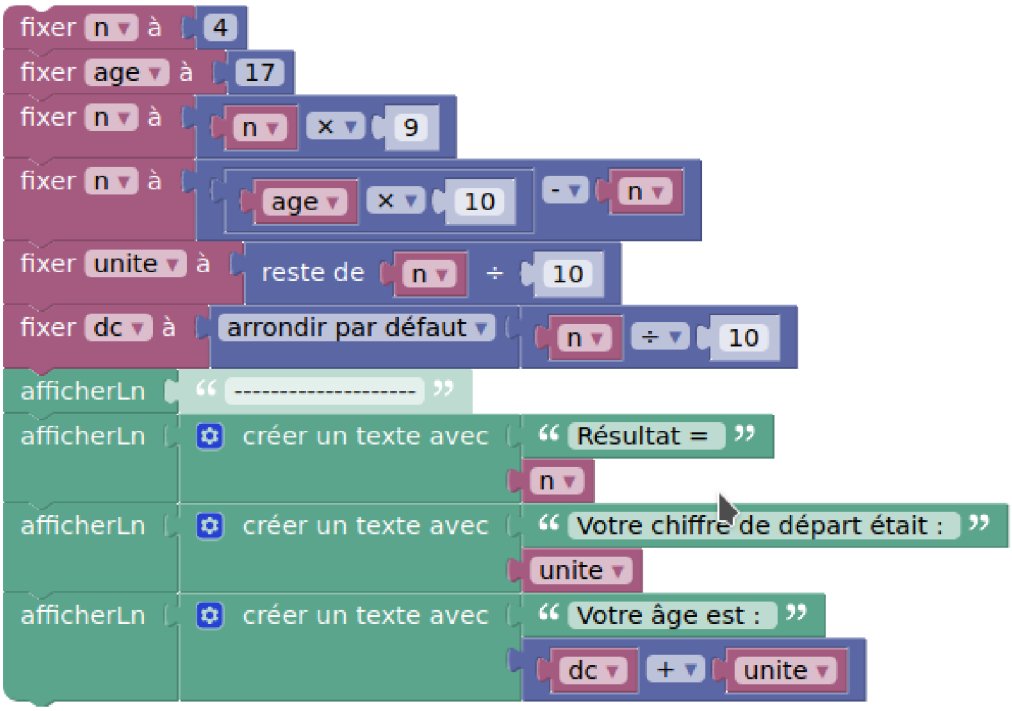
La variable "n" est le nombre choisi au point a).
"2001" est la date de naissance.
L'algorithme doit être adapté chaque année, l'année prochaine, il faudra soustraire 2020 au lieu de 2019.

Voici un autre tour de magie, à programmer.
*) Le tour de magie : "Quel est votre âge ?"
° Pensez à un chiffre entre 1 et 9.
° Multipliez ce chiffre par 9.
° À 10 fois votre âge, soustrayez le résultat obtenu.

-> Le chiffre des unités est le chiffre auquel vous avez pensé au départ.
-> Ajoutez le chiffre des unités au nombre constitué des autres chiffres, cela donnera votre âge.

Exemples :
C = 8, âge = A = 17
Résultat = 10 * 17 - 9 * 8 = 98
-> Le chiffre des unités vaut bien C
-> Le chiffre des unités plus celui des dizaines vaut 8 + 9 = 17 = l'âge.

C = 3, âge = A = 17
Résultat = 10 * 17 - 9 * 3 = 143
-> Le chiffre des unités vaut bien C
-> Le chiffre des unités plus celui formé par les autres chiffres vaut 3 + 14 = 17 = l'âge.

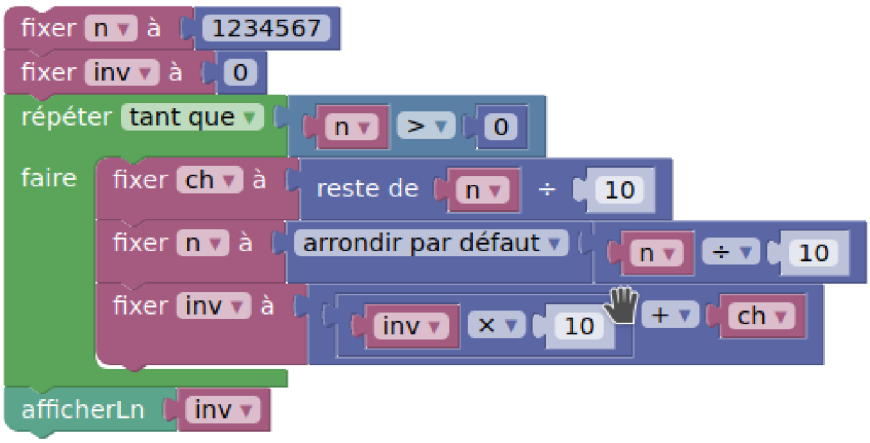


La variable "n" est le nombre pensé au départ.
17 est votre âge.

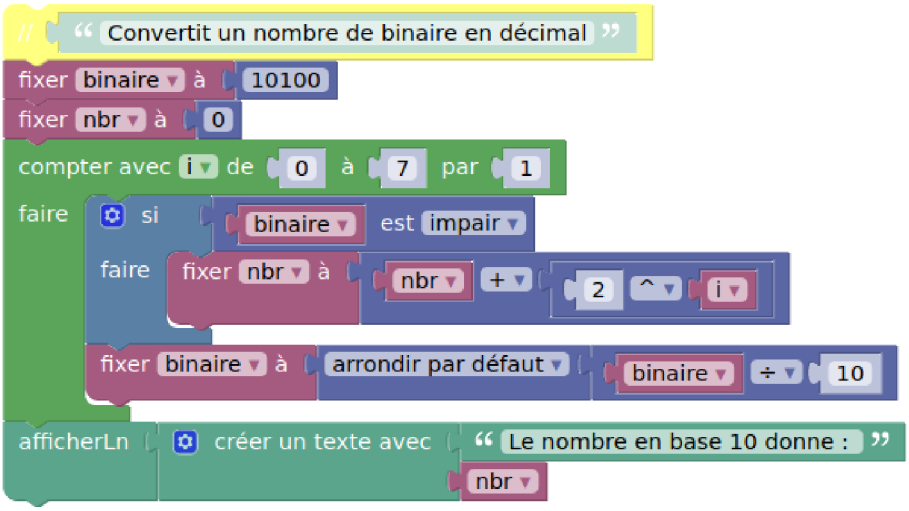
*) Vous avez déjà abordé le binaire.

a) Saurez-vous écrire un algorithme, qui pour un nombre entier positif donné, affiche ce nombre à l'envers.
Exemple : donnée = 1234567 ; résultat affiché = 7654321.

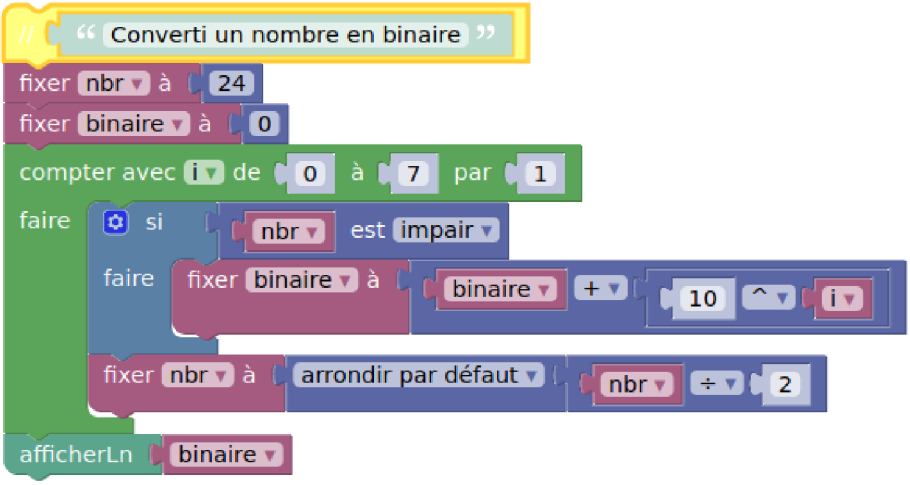
b) Comme l'algorithme précédent, mais au lieu d'afficher le nombre à l'envers, stocke le nombre à l'envers dans une nouvelle variable.



c) Saurez-vous écrire un algorithme, qui pour un nombre donné en binaire, crée un nombre correspondant à la conversion en base 10 du nombre donné.
Exemple : donnée = 1101 ; résultat = 13.



d) Saurez-vous écrire un algorithme, qui pour un nombre donné en base dix, crée un nombre correspondant à la conversion en binaire du nombre donné.
Exemple : donnée = 20 ; résultat = 10100.



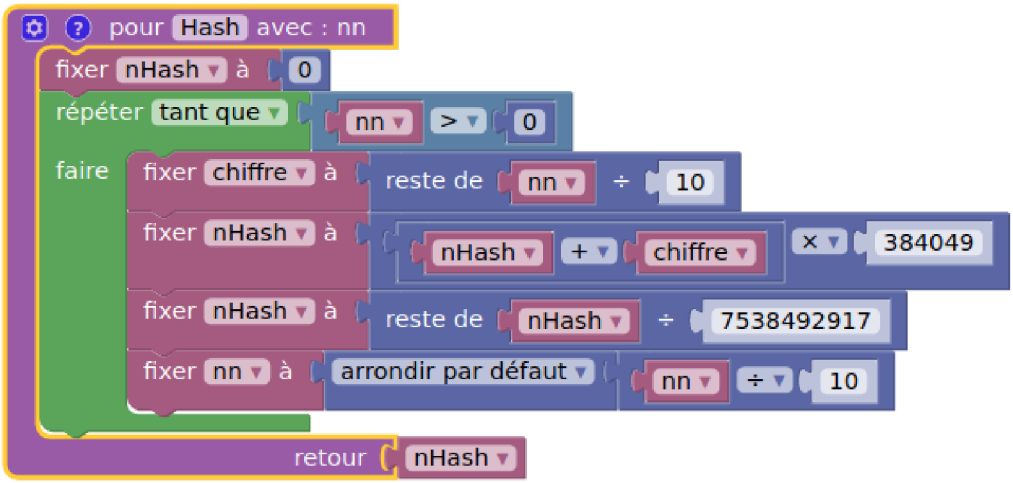
Un exemple de fonction de hachage, donné sous forme d'algorithme.
Un **algorithme** est une suite d'instructions.

Soit *n* le nombre à "hacher".
Suite d'instructions dans un pseudo code :
hash = 0
tant que n > 0 faire :
    chiffre = reste de la division de n par 10
    hash = (hash + chiffre) * 384049
    hash = reste de la division de hash par 7538492917
    n = arrondi vers le bas de (n / 10)

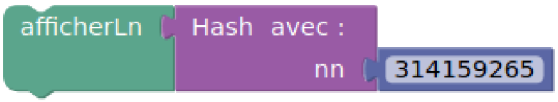
retourner hash ; *donc* hash = *Hash (n)*.

Exercice 1
Allez dans : http://www.juggling.ch/gisin/program/blockly/abgex/ex2000/ex2000.html
Configuration > Divers exemples, au choix... > Choix 3

° Programmez une fonction "Hash(n)", selon l'algorithme précédent qui calcule la fonction de hachage du nombre n.



° Tester votre fonction sur : n = 314'159'265 vous devez obtenir Hash (n) = 3062988386



° Comparez votre fonction avec celle d'autres élèves.

° Trouvez un autre nombre n_2 tel que sa fonction de hachage Hash (n_2) = 3'062'988'386
La fonction de hachage est faite pour qu'il soit très difficile que deux nombres aient le même hashcoding.
Dans la situation de l'exercice, qui a été simplifiée, il est possible de trouver un deuxième nombre, par exemple n_2 = 2'247'241'369. (Un autre évident : n_2 = 3141592650)
Dans la situation réelle, avec la fonction SHA256, personne n'a réussi a trouver deux nombres qui ont le même hashcoding par cette fonction. Donc cette signature est très très difficilement falsifiable.