

Salut tous les deux,

Bon alors cet exo est simple une fois qu'il est expliqué correctement, par contre il est aussi super important dans le sens où la base de la programmation tu l'as là : le pseudo-code.

J'ai fait court, le pseudocode comme t'as du le voir en cours c'est globalement ce que tu veux que ton programme fasse mais en l'écrivant encore en 'bon français'. C'est un peu comme si tu devais l'expliquer à un collègue un peu con, faut lui donner étape par étape ce qu'il doit faire. Une sorte de recette de cuisine simpliste.

Important: *si ton pseudocode est bon, tu gagnes la race de temps par la suite, t'auras moins de problèmes à le traduire en vrai code et tes cheveux sentiront bon la lavande!*

Donc dans ton cas là on te demande de 'penser' un programme qui fasse que l'utilisateur entre une suite d'entiers positifs par ordre croissant.

Regle #1 : Fais au plus simple.

Regle #2 : Fais le petit à petit.

Regle #3 : Écrire le programme comme si l'utilisateur est un génie, puis le transformer pour que même un chimpanzé débile avec une lobotomie frontale puisse l'utiliser (et d'hab l'utilisateur final est même encore plus con que cela)

Concrètement, faut déjà que *ton programme se lance*, ensuite tu demandes à l'utilisateur de rentrer une valeur, tu la gardes dans une liste, pour qu'il puisse rentrer plusieurs valeurs on va lui proposer de rentrer une valeur (boucle) et le programme quitte.

En pas tout à fait pseudocode:

Demarrage programme:

Afficher nom du programme et auteure

Fais une boucle à partir d'ici

Demander la saisie de l'utilisateur [Entrer des entiers positifs croissants:]

Ajouter nouvelle valeur à la liste des valeurs précédentes

Fin de boucle

Afficher annonce de fin de programme [Merci d'avoir joué :)]

Fin du programme.

Une fois que tu es dans la boucle, rien ne t'en fait sortir elle continue à l'infini (dans la plupart des cas c'est pas une bonne chose, ton ordi va mouliner dans le vide) il faut donc lui donner une "condition" pour arrêter de boucler. C'est donné dans la 2ème consigne : **"Si la valeur donnée est zéro, l'utilisateur a fini sa saisie (afficher la liste de valeurs)"**

Donc on modifie le code d'au dessus :

Demarrage programme:

Afficher nom du programme et auteure

Boucle tant que la valeur entree n'est pas zéro

Demander la saisie de l'utilisateur [Entrer des entiers positifs croissants:]

Si le nombre entré est égal a zéro:

afficher la liste des valeurs

quitter le programme(Succès!Merci d'avoir joué :))

Autrement :

Ajouter nouvelle valeur a la liste des valeurs précédentes

Fin de boucle

Fin du programme.

Ok pas mal. Regarde bien OU on ecrit la condition "**Si**" on le fait AVANT d'ajouter la valeur a la liste, histoire de pas coller un zéro qui veut dire fin de programme dans notre liste de valeurs. Cela dépend bien sur des cas que tu traites.

Egalement quand tu utilise une condition « **Si** » il faut dans 99.99% des cas dire au programme quoi faire si la condition ne se réalise pas. C'est mon « **Autrement** ». En gros quand tu programme des choses compliquées cela permet si jamais tu as pas prévu certaines conditions de pas tout faire bugger.

Mtn il faut s'occuper de la premiere consigne: "**Si le dernier entier est inferieur ou egal a l'entier précédent, quitter le programme et afficher un message d'erreur**"

Demarrage programme:

Afficher nom du programme et auteure

Boucle tant que la valeur entree n'est pas zéro

Demander la saisie de l'utilisateur [Entrer des entiers positifs croissants:]

Si le nombre entré est égal a zéro:

afficher la liste des valeurs

quitter le programme(Succès!Merci d'avoir joué :))

Si le nombre entré est inférieur ou égal au précédent:

quitter le programme(Echec ! On avait dit des valeurs croissantes !)

Autrement :

Ajouter nouvelle valeur a la liste des valeurs précédentes

Fin de boucle

Fin du programme.

Ici si tu places ton 2eme « **Si** » AVANT l'autre, le « **Si le nombre entré est égal a zéro:** » ne sera jamais exécuté ! En effet comme tu travailles avec des nombres entiers positifs, 0 sera toujours inférieur ou égal a ta valeur. Du coup ton programme fonctionnera pas correctement.

Suffit maintenant de retraduire cela en pseudocode, c'est-à-dire sans mettre autant de blabla et en le faisant plus ressembler a quelque chose de plus proche d'un programme :

n'est pas égal à ça s'écrit :!=

inférieur ou égal : <=

supérieur ou égal : >=

Debut:

Affiche(nom du programme, auteure)

Tant que valeur != 0

Affiche(« Entrer des entiers positifs croissants: »)

Saisie de l'utilisateur --> stocker dans **valeur**

Si valeur = 0 :

Affiche **liste_des_valeurs**

Quitte(Succès!Merci d'avoir joué :))

Si valeur <= valeur_précédente :

Quitte(Echec ! On avait dit des valeurs croissantes !)

Autrement :

Ajouter **valeur** a la **liste des valeurs**

Fin boucle

Fin.

Encore un petit effort.

Comme jte disais un utilisateur faut le voir comme ayant un QI de banane pas épluchée. Rien ne l'empêche de rentrer une lettre par exemple. Comme tu le vois dans notre prog ce cas n'est pas traité. Il suffit d'ajouter une condition pour cela « il faut que ce soit un entier positif ».

Ensuite, c'est l'utilisateur qui nous donne « **valeur** » mais comme trouver **valeur_précédente** ?

Il suffit de garder en mémoire la valeur_précédente juste avant ou juste après avoir ajoutée la valeur saisie par l'utilisateur a la liste !

Debut:

Affiche(nom du programme, auteure)

Tant que valeur_saisie != 0

Affiche(« Entrer des entiers positifs croissants: »)

Saisie de l'utilisateur --> stocker dans **valeur_saisie**

Si valeur_saisie != entier positif :

Quitte(Echec ! La valeur saisie n'est pas un entier positif !)

Si valeur_saisie = 0 :

Affiche **liste_des_valeurs**

Quitte(Succès!Merci d'avoir joué :))

Si valeur_saisie <= valeur_précédente :

Quitte(Echec ! On avait dit des valeurs croissantes!)

Autrement :

Ajouter **valeur_saisie** a la **liste des valeurs**

valeur_précédente = valeur_saisie

Fin boucle

Fin.

Comme tu le vois, on a deux conditions qui mènent a un echec du programme (si pas croissant OU si pas un nombre), on peut les regrouper c'est plus propre :

Debut:

Affiche(nom du programme, auteure)

Tant que `valeur_saisie != 0`

Affiche(« Entrer des entiers positifs croissants: »)

Saisie de l'utilisateur --> stocker dans `valeur_saisie`

Si `valeur_saisie != entier positif OU valeur_saisie <= valeur_précédente :`

Quitte(Echec ! La valeur saisie n'est pas un entier positif croissant !)

Autrement Si `valeur_saisie = 0 :`

Affiche `liste_des_valeurs`

Quitte(Succès!Merci d'avoir joué :))

Autrement :

Ajouter `valeur_saisie` a la `liste des valeurs`

`valeur_précédente = valeur_saisie`

Fin boucle

Fin.

Du coup, ce qu'il se passe c'est qu'on regarde avec « Si » que la valeur est valide (un entier positif croissant), « Autrement Si » vérifie que c'est pas égal a zéro qui est notre valeur de sortie de programme, et « Autrement » c'est le cas idéal on ajoute la valeur a notre liste .

Il n'ya pas de vrais « règles » pour le pseudocode, chacun fait un peu comme il veut, mais ça peut etre utile d'utiliser directement les mots-clés en anglais, ce qui donne :

Start:

Affiche(nom du programme, auteure)

While `valeur_saisie != 0`

Affiche(« Entrer des entiers positifs croissants: »)

Saisie de l'utilisateur --> stocker dans `valeur_saisie`

If `valeur_saisie != entier positif or valeur_saisie <= valeur_précédente :`

Quitte(Echec ! La valeur saisie n'est pas un entier positif croissant!)

Else If `valeur_saisie = 0 :`

Affiche `liste_des_valeurs`

Quitte(Succès!Merci d'avoir joué :))

Else :

Ajouter `valeur_saisie` a la `liste des valeurs`

`valeur_précédente = valeur_saisie`

Fin boucle

End.

Et voilà ! La ton programme va faire ce qu'on lui demande et si l'utilisateur fait le con, le programme l'engueulera :)

Je sais que c'est un peu lourd quand tu débutes mais si tu le fais bien comme cela, pas a pas ca ira de plus en plus vite.

Si t'as des questions hésites pas, va faire un tour sur le « site du zéro » y a des cours de programmation pour les très débutants.

Bon courage !