

TP 1

Objectif du TP : résoudre des algorithmes pour répondre à un problème métier à l'aide du logiciel algobox.

Contexte global

Vous êtes une équipe de 3 développeurs et vous travaillez pour une nouvelle banque en ligne la GTM Bank.

Votre employeur souhaite créer un programme qui permet d'effectuer des opérations bancaires.

Certains clients auront droit à un découvert, cependant la banque applique des pénalités lorsque le client utilise une partie ou l'intégralité de son découvert autorisé.

Une fois le montant crédité sur le compte, le client peut effectuer autant de retraits qu'il souhaite jusqu'à l'épuisement de son solde ou de l'utilisation totale de son découvert.

- Le retrait est autorisé uniquement lorsque le solde est suffisant.
- Si le client n'a pas opté pour le découvert, le nouveau solde ne doit pas être négatif.
- Si le client a opté pour le découvert, le nouveau solde ne doit pas dépasser le montant du découvert autorisé.
- Le montant du découvert est toujours positif.
- Le montant du retrait est toujours positif et d'un montant minimum d'un euro.

Consigne

1. Pour cette partie, chacun travaille de son côté
2. Vous travaillez tous sur le même fichier que vous nommerez comme bon vous semble.
3. Écrivez votre algorithme sans utiliser les fonctions.
4. Votre programme fonctionne indépendamment des autres.
5. Chacun à une mission à accomplir mais vous travaillez en équipe,entraidez-vous

Le premier développeur est responsable de l'ouverture de compte, le second des opérations de retrait et le dernier du calcul des agios.

Mission du développeur 1 : ouverture du compte

Écrivez un algorithme qui demande au client lors de la création d'un nouveau compte bancaire chez GTM Bank :

1. La possibilité d'avoir recours au découvert (oui ou non). Si oui, le montant du découvert autorisé entre 100 et 2000 €, le client détermine ce montant en fonction de ses besoins.
2. Le montant à transférer pour l'ouverture du compte, la somme initiale doit être au minimum de 500 €
3. Forcez le client à respecter les conditions (montant du départ et la fourchette du découvert). Si le client ne respecte pas une de ces conditions, un message d'erreur apparaît "Vous ne respectez pas les conditions d'ouverture du compte"

Pour résumer :

- Demandez au client s'il souhaite avoir droit au découvert.
- Si oui, demandez au client le montant du découvert (valeur comprise entre 100 et 2000 €).
- Demandez au client le montant à transférer sur le compte (≥ 500 €) pour l'ouverture du compte.
- **En sortie**, votre algorithme affiche la somme initiale et le montant du découvert (zéro s'il n'a pas opté pour le découvert), ou un message d'erreur

Mission du développeur 2 : retraits

Écrivez un algorithme qui prend **en entrée** :

1. Le montant de son découvert autorisé (si le client n'a pas choisi d'avoir droit au découvert, il doit saisir zéro).
2. Le solde du compte en cours.
3. Le montant du retrait souhaité par le client.

Attention le montant du solde s'il est négatif ne doit pas être supérieur en valeur absolue au montant du découvert.

Exemple :

- Non autorisé, découvert de 500 € et solde de -700 €.
- Autorisées, découvert de 500 € et solde de -500 € ou découvert de 900 € et solde de -300 €.

Si le solde est suffisant, votre programme doit demander au client s'il souhaite refaire un nouveau retrait une fois.

Le client peut arrêter les opérations en saisissant la valeur 0 (ou un autre caractère alphanumérique).

En sortie, votre algorithme affiche :

1. Le nouveau solde ou un message "solde insuffisant"
2. Le montant du découvert

Pour résumer :

- Demandez le montant du découvert (0 s'il n'y a pas découvert autorisé).
- Demandez le solde en cours.
- Si le solde le permet, demandez au client s'il souhaite effectuer un autre retrait une fois. Il saisira la valeur du retrait ou 0 pour quitter
- Affichez le nouveau solde et le montant du découvert ou "solde insuffisant"

Mission du développeur 3 : agios

Écrivez un algorithme qui calcul les agios avec un taux de 10% pour un découvert utilisé durant X jours.

Par exemple, j'ai utilisé 500 euros de mon découvert autorisé durant 15 jours, j'aurai une pénalité à payer à la banque de 2.05 € qui correspond à l'opération suivante : $(500 * 15 * (10/100)) / 365$ $\Leftrightarrow 500 * 15 * 0.1 / 365$

En entrée, votre algorithme prend :

1. Le montant du découvert.
2. Durée d'utilisation du découvert (en jour)

En sortie, votre algorithme affiche la somme que le client devra payer à la banque

****arrondie à 2 chiffres après la virgule.****

Votre algorithme s'arrête immédiatement lorsque le client n'a pas droit au découvert (montant du découvert égale à zéro), affichez le message "Découvert non autorisé => pas d'agios".

Forcez le client à saisir les bonnes valeurs, sinon affichez un message d'erreur "Valeurs incorrectes" :

- Montant du découvert compris entre 100 € et 2000 € ou égale à zéro lorsqu'il n'a pas droit au découvert.
- Le nombre de jours compris entre 1 et 365.

Formulaire pour calculer d'agios

`AgiOS = (montant utilisé du découvert * nombre de jour d'utilisation * taux de la banque) / 365`

`Le taux de banque = 10 / 100 = 0.1`

Pour arrondir

Utilisez la fonction d'arrondie d'AlgoBox `ALGOBOX_ARRONDIR(agios,2)`

Outils en ligne pour vérifier le résultat du calcul d'agios

<https://www.coover.fr/outils/calcul-agios>

Travail collectif de fin :

Créer l'algorithme qui prend en compte ces trois parties.

Il demande dans un premier lieu l'ouverture d'un compte.

On peut ensuite saisir 1 ou 2 pour effectuer un retrait ou un calcul d'agios.