TP 02

Structures algorithmiques

Lien Capytale https://capytale2.ac-paris.fr/web/c/f807-628160/mcer

Savoirs et compétences :

❒ .

Structures algorithmiques

Structures conditionnelles

Question 1 *Écrire une fonction* `val_absolue(x:float) -> float` *prenant un paramètre flottant* `x` *et retournant la valeur absolue de x (sans utiliser la fonction* `abs`*).*

Une agence de voyage propose un voyage organisé où l'on peut s'inscrire en groupe. Le prix par personne est dégressif selon le nombre de personnes dans le groupe :

- 80 euros (par personne) pour un groupe d'une ou deux personnes ;
- 70 euros (par personne) pour un groupe de 3 à 5 personnes ;
- 60 euros (par personne) pour un groupe de 6 à 9 personnes ;
- 50 euros (par personne) à partir de 10 personnes.

On appelle n la variable contenant le nombre de personnes dans le groupe.

Question 2 *Écrire une fonction* `cout_voyage(n:int)->int` *qui renvoie le prix total pour l'ensemble du groupe.*

Question 3 *Écrire une fonction* `compter_for(n:int)->None` *qui affiche successivement tous les nombres de 0 inclus à n exclus (boucle* `for`*).*

Question 4 *Tester le bon fonctionnement de votre fonction.*

Question 5 *Écrire une fonction* `compter_while(n:int)->None` *qui affiche successivement tous les nombres de 0 inclus à n exclus (boucle* `while`*).*

Question 6 *Tester le bon fonctionnement de votre fonction.*

Question 7 *Écrire une fonction* `compter_rebours_for(n:int)->None` *qui affiche successivement tous les nombres de n inclus à 0 inclus (boucle* `for`*).*

Question 8 *Tester le bon fonctionnement de votre fonction.*

Question 9 *Écrire une fonction* `compter_rebours_while(n:)->None` *qui affiche successivement tous les nombres de n inclus à 0 inclus (boucle* `while`*).*

Question 10 *Tester le bon fonctionnement de votre fonction.*

Question 11 *Écrire une fonction* `epeler(mot:str)->None` *qui affiche successivement toutes les lettres du mot* `mot`.

Question 12 *Tester le bon fonctionnement de votre fonction.*

Répétition conditionnelle

Question 13 *Déterminer le plus petit entier n tel que* $1+2+...+n$ *dépasse strictement N. Pour cela on implémentera la fonction* `plus_petit_entier(N:int)->int`.

Initiation à l'utilisation des listes

Applications directes

Question 14 *Implémenter une fonction compter(n :int) -> list renvoyant la liste des entiers de 1 à n inclus.*

Question 15 *Vérifier votre fonction en vous appuyant sur un exemple.*

Question 16 *Implémenter une fonction* `compter_pairs(n)` *renvoyant la liste des entiers pairs de 0 à n inclus.*

Question 17 *Vérifier votre fonction en vous appuyant sur un exemple.*

Question 18 *Implémenter une fonction* `compter_impairs(n)` *renvoyant la liste des entiers impairs de 1 à n inclus.*

Question 19 *Vérifier votre fonction en vous appuyant sur un exemple.*

Simulation d'un prêt bancaire

Objectif L'objectif de ce TP est de comprendre comment sont calculées les mensualités d'un prêt bancaire à taux fixe. Vous souhaitez emprunter 100 000€ à la banque. La banque vous propose un taux annuel de 3% sur 10 ans. Que cela signifie-t-il ?

Cas 1 : remboursement annuel

On se place dans la situation où vous avez la possibilité de faire un remboursement à la banque par an. Pour se rétribuer, la banque vous fait donc payer, chaque année, 3% de la somme restant à rembourser. C'est cette somme qu'on appelle les intérêts. Ainsi, la première année, il faudrait payer 3000 € d'intérêts.

On cherche à savoir le montant annuel à rembourser à la banque.

Appelons :

- t le taux annuel du prêt ;
- C_0 le montant initial emprunté ;
- C_i le montant restant à rembourser à la fin de l'année i ;
- A l'annuité qui se définit par le montant (inconnu) à rembourser chaque année ;
- I_i le montant des intérêts à rembourser à la fin de l'année i ;
- N la durée du prêt en années.

À la fin de la première année :

- $I_1 = C_0 \times t$
- $C_1 = C_0 - (A - I_1) = C_0 - A + I_1 = C_0 - A + C_0 \times t = C_0(1+t) - A$.

Remarque : A doit être supérieur à $C_0 \times t$.

À la fin de la deuxième année :

- $I_2 = C_1 \times t = C_0(1+t)t - At$
- $C_2 = C_1 - (A - I_2) = C_0(1+t) - A - A + C_0(1+t)t - At = C_0(1+t)^2 - A(2+t)$.

Aini, au bout de la ième année,

- $I_i = C_{i-1}\,t$;
- $C_i = C_{i-1}(1+t) - A$.

On montre que $A = \frac{C_0 t(1+t)^N}{(1+t)^N - 1}$.

Question 20 *Implémenter la fonction* `calcul_annuite(C0:float,t:float,N:int)` *permettant de déterminer le montant d'une annuité.*

Question 21 *Vérifier que dans le cas de l'exemple donné, les annuités s'élèvent à 11 723,05€.*

Question 22 *Implémenter une fonction* `reste_a_rembourser(C0,t,a,n)` *où* `C0` *est le montant prêté,* `t` *le taux annuel du prêt,* `a` *le montant d'une annuité,* `n` *le nombre d'annuités. Cette fonction calculera le montant à rembourser après le versement de la nième annuité.*

Question 23 *Vérifier qu'au bout de 10 ans, la somme globale a été payée.*

Question 24 *Écrire une fonction* `cout_total(C0,t,N)` *renvoyant le coût total du crédit, c'est-à-dire le total de ce que vous avez payé moins le montant du prêt.*

Question 25 *Vérifier que le cout total du prêt est de 17 230,50 €.*

Cas 2 : remboursement mensuel

Question 26 *Donner le montant des mensualités du prêt.*

Question 27 *Quel sera le reste à payer au bout de 10 ans ?*

Question 28 *Déterminer le coût total du prêt.*

Tracer de courbes

Question 29 *Dans le cas d'un remboursement annuel, écrire une fonction interet(C0 :float,t :float,N :int) -> list renvoyant la liste des intérêts payés chaque année.*

Question 30 *Dans le cas d'un remboursement annuel, écrire une fonction capital(C0 :float,t :float,N :int) -> list renvoyant la liste du capital restant à rembourser en fin de chaque année.*