

# Examen

## Corrigé

**Remarques préliminaires :** Cet examen est d'une durée de 2 heures. Les documents de la formation PAD sont autorisés ainsi que les notes manuscrites. Les calculatrices, téléphones ou autres appareils ne sont pas autorisés.

Pour tous les sous-programmes, il est demandé de donner :

1. la signature du sous-programme en *langage algorithmique* en précisant bien s'il s'agit d'une procédure ou d'une fonction d'une part, et les modes de passage des paramètres (*in*, *out* ou *in out*) d'autre part ;
2. les premiers niveaux de raffinement s'ils sont différents du code ;
3. le code, en Algorithmique ou dans le langage étudié (C, VBA, F ou Python), avec les commentaires liés au raffinement.

Sauf demande contraire explicite, aucun sous-programme ne doit être interactif (donc ni affichage, ni saisie).

D'autres sous-programmes que ceux explicitement demandés peuvent être définis.

**Barème indicatif :** Exercice 1 : 4 pts, Exercice 2 : 16 pts (3 + 2 + 2 + 2 + 2 + 5)

Exercice 1 : Les offres .....	3
Exercice 2 : Les projets .....	4

Nous nous intéressons ici à une plateforme de financement participatif. Elle permet à des entreprises de faire un emprunt auprès de prêteurs. Elle utilise un système d'enchère inversée : les prêteurs choisissent le taux et seules les offres au taux le plus bas sont retenues. La figure 1 présente une capture d'écran du site Web<sup>1</sup> d'un projet en demande de financement.

Une entreprise dépose une offre d'emprunt correspondant à un projet caractérisé par le nom de l'entreprise (Ouroboros sur l'exemple), le montant souhaité (150 000 euros), la durée en nombre de mois (de 6 à 60 mois, 36 mois pour Ouroboros) et la date limite pour faire une offre de prêt. Il y a bien sûr une description de l'entreprise et du projet que l'emprunt permettra de réaliser.

Jusqu'à la date limite, un prêteur peut faire une offre en choisissant un taux d'intérêt et un montant. Le taux doit être compris entre 4% et 10% au dixième de pourcent près. Le montant doit être de 20 euros minimum. Toutes les offres de prêts sont identifiées par leur rang (1 pour la première, 2 pour la deuxième, etc.).

Si de nouveaux prêteurs font des offres plus intéressantes, les offres avec un taux plus élevé sont éliminées (ceci est détaillé à la question 2.6). Ainsi, une offre de prêt est marquée soit comme "Non retenue", soit "En cours". Par exemple, la première offre, 40 € à 9% est toujours en cours alors que le deuxième à 10% a été éliminée par des offres à un taux plus faible.

1. La présentation des offres de prêt a changé en mai 2016 et cette interface n'est plus disponible.

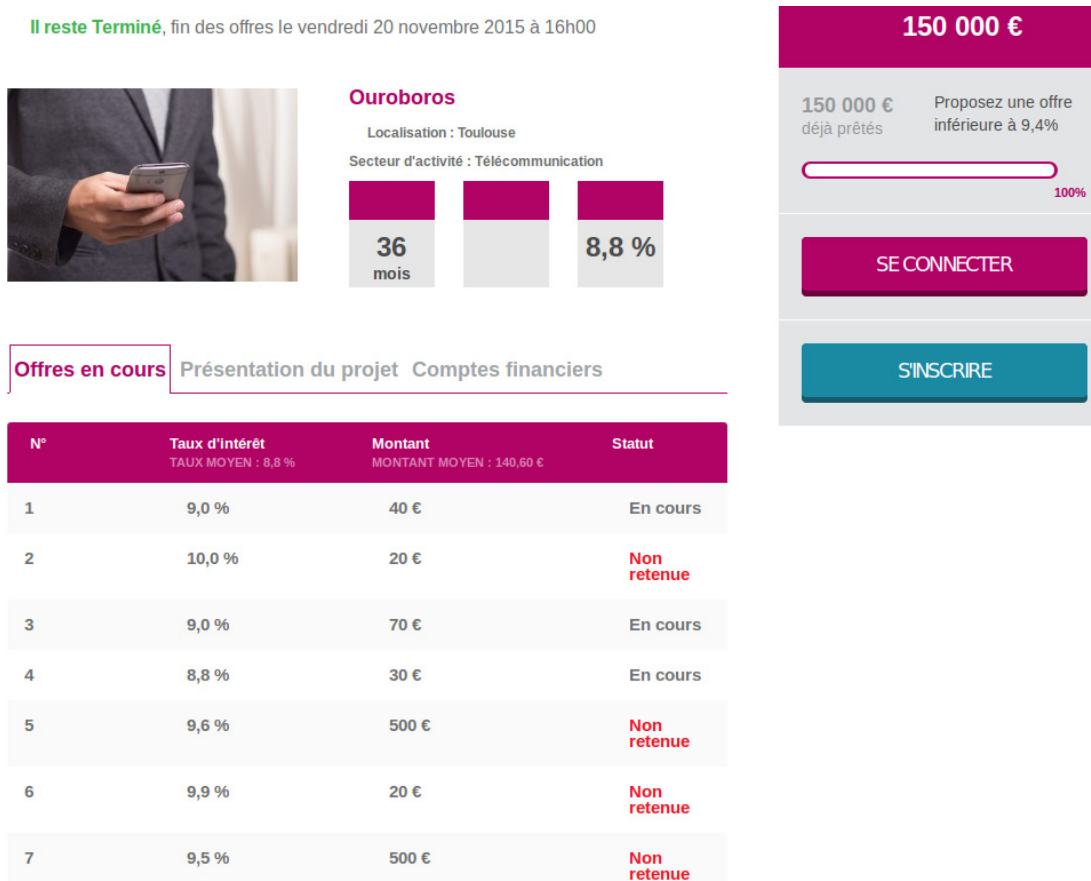


FIGURE 1 – Exemple de projet qui vient juste de se terminer. La société Ouroboros emprunte 150 000 euros sur 36 mois. Le taux obtenu (moyenne du taux des offres) est 8,8%. Seules les 7 premières offres de prêts apparaissent. On note en haut à droite qu'à la fin des enchères, les prêteurs pouvaient proposer un taux inférieur à 9.4%. Ceci signifie que toutes les offres proposées à un taux strictement inférieur à 9,4% sont en cours.

À la date limite, si les prêteurs ont réussi à atteindre le montant demandé par l'entreprise le prêt est accordé en utilisant les offres qui ont proposé le taux le plus bas (enchère inversée). Si le montant n'est pas atteint, l'emprunt ne se réalise pas et les prêteurs récupèrent leur mise qu'ils peuvent proposer à d'autres projets.

### Exercice 1 : Les offres

Pour une offre, on se limite à son montant (20 € minimum), son taux (entre 4 et 10 %) et son état ("En cours" ou "Non retenue").

Définir le type Offre (et éventuellement d'autres types utiles) ainsi que le sous-programme qui permet d'initialiser une offre à partir de son montant et de son taux. L'offre sera dans l'état "en cours".

### Solution :

```
1  Type
2      Etat = (EN_COURS, NON_RETENUE)
3
4      Offre =
5          Enregistrement
6              montant: Entier -- en euros
7              taux: Entier      -- en millième
8              etat: Etat
9              -- Invariant :
10             -- montant >= 20
11             -- taux >= 40 && taux <= 100
12          FinEnregistrement
```

```

1 Procédure initialiser_offre(offre: out Offre; montant: in Entier; taux: Entier)
2   -- Initialiser l'offre à partir de son montant et de son taux.
3   --
4   -- Nécessite
5   --   taux >= 40 && taux <= 100
6   --   montant => 20
7   --   montant Mod 1000 = 0   -- multiple de 1000 (non dans le sujet)
8   Début
9     offre.montant <- montant
10    offre.taux <- taux
11    offre.etat <- EN_COURS
12   Fin

```

## Exercice 2 : Les projets

Un projet est caractérisé par le montant demandé (en euros), sa durée (en mois) et les offres de prêt formulées. Nous nous intéressons ici à quelques opérations sur un projet, en particulier le total des offres de prêt faites et le reste à financer. Ces informations sont illustrées sur l'exemple de la figure 2 (qui est composé de 11 offres pour un projet de 10 000 €)

rang	montant	taux	total	reste
1	200	7,0	200	9800
2	1000	10,0	1200	8800
3	500	8,0	1700	8300
4	800	4,0	2500	7500
5	1000	9,9	3500	6500
6	300	10,0	3800	6200
7	2000	7,0	5800	4200
8	1400	9,0	7200	2800
9	1500	9,0	8700	1300
10	1000	8,0	9700	300
11	900	9,5	10600	0

FIGURE 2 – Exemple d'offres sur un projet de 10 000 €. Les deux dernières colonnes sont des informations calculées : la somme totale des offres de prêt et le reste à financer pour le projet.

**2.1** Définir le type `Projet` ainsi qu'un sous-programme pour l'initialiser à partir du montant demandé et de la durée. Quand le projet est initialisé, il n'y a encore aucune offre de prêt. Les offres seront conservées dans un tableau et l'indice dans le tableau correspondra au rang de l'offre.

**Solution :**

```

1  Constante
2      MAX = 100000    -- arbitraire
3
4  Type
5      Projet =
6          Enregistrement
7              montant: Entier -- en euros
8              duree: Entier   -- en mois
9              offres: Tableau [MAX] De Offre
10             nb_offres: Entier -- nombre d'offres faites sur ce projet
11         FinEnregistrement

1  Procédure initialiser_projet(projet: out Projet; montant: in Entier; durée: in Entier)
2      -- Initialiser un projet à partir de son montant et de sa durée.
3      -- Initialement le projet n'a pas encore d'offre.
4      Début
5          projet.montant <- montant
6          projet.durée <- durée
7          projet.nb_offres <- 0
8      Fin

```

2.2 Écrire un sous-programme qui calcule la somme de toutes les offres de prêt faites sur un projet. Sur la figure 2, c'est la valeur de la colonne "total" en considérant que la ligne correspond à la dernière offre faite sur le projet. Par exemple, au moment où le projet reçoit la 6<sup>e</sup> offre (il y a donc 6 offres de prêt), le total des offres de prêt correspond à la valeur 3 800.

**Solution :**

```

1  Fonction somme_offres(projet: in Projet): Entier
2      -- Somme des montants de toute les offres faites sur un projet.
3      Début
4          Résultat <- 0
5          Pour i <- 0 Jusqu'à i = nb_offres -1 Faire
6              Résultat <- Résultat + projet.offres[i].montant
7          FinPour
8      Fin

```

2.3 Écrire un sous-programme qui indique pour un projet donné le reste à financer. Sur la figure 2, la valeur de la colonne "reste" correspond au reste à financer en considérant que la ligne correspond à la dernière offre faite. Ainsi, après la 6<sup>e</sup> offre, le reste à financer correspond à la valeur 6 200.

**Solution :**

```

1  Fonction reste_a_financer(projet: in Projet): Entier
2      -- Reste à financer sur un projet.
3      Var
4          somme: Entier
5      Début
6          somme <- somme_offres(projet)
7          Si somme >= projet.montant Alors
8              Résultat <- 0
9          Sinon
10             Résultat <- projet.montant - somme
11         FinSi
12     Fin

```

2.4 Écrire un sous-programme qui calcule, sur un projet donné, la somme des montants de toutes les offres qui ont un taux d'intérêt inférieur ou égal à un taux donné. Sur l'exemple de la figure 2, la somme des prêts dont le taux est inférieur ou égal à 7% est 3 000 € (offres de rang 1, 4, et

7). Si on considère le taux de 10%, le résultat est 10 600 (toutes les offres). Pour le taux 4%, on obtient 800 (seulement l'offre de rang 4).

**Solution :**

```

1  Fonction somme_offres_inferieures_taux(projet: in Projet; taux: in Entier): Entier
2      -- Somme des montants de toute les offres faites sur un projet dont le
3      -- taux est inférieur à un taux donné.
4      Début
5          Résultat <- 0
6          Pour i <- 0 Jusqu'à i = nb_offres -1 Faire
7              Si projet.offres[i].taux <= taux Alors
8                  Résultat <- Résultat + projet.offres[i].montant
9              FinSi
10         FinPour
11     Fin

```

**2.5** Une compagnie d'assurance qui soutient la plateforme s'est engagée à prêter sur certains projets. Quand elle prête, elle fait une offre de prêt qui représente exactement 20% du financement demandé.

Écrire un sous-programme qui calcule le rang de la première offre dont le montant est exactement égal à 20% du financement demandé. Si une telle offre n'existe pas, le résultat doit être -1. Sur l'exemple de la figure 2, la compagnie s'est positionnée car l'offre de rang 7 à 2 000 € a bien un montant qui représente 20% des 10 000 € du projet.

**Solution :**

```

1  Fonction rang_offre_20(projet: in Projet): Entier
2      -- Rang de la première offre dont le montant représente 20% du montant
3      -- à financer sur le projet. -1 si aucune offre à 20%.
4      Variable
5          montant_cherché: Entier
6      Début
7          montant_cherché <- projet.montant * 20 / 100
8          Résultat <- 0
9          TantQue Résultat < projet.nb_offres -- encore des offres à considérer
10             Et projet.offres[Résultat].montant <> montant_cherché -- pas l'offre cherché
11         Faire
12             Résultat <- Résultat + 1
13         FinTQ
14         Si Résultat >= projet.nb_offres Alors Faire
15             Résultat <- -1
16         FinSi
17     Fin

```

**2.6** Dans cette question on ne demande pas d'écrire le code mais simplement de donner les premiers niveaux de raffinage.

On s'intéresse maintenant au sous-programme qui permet de faire une nouvelle offre de prêt sur un projet.

Sur l'exemple de la figure 2, les offres 1 à 10 sont en cours car la somme de leur montant est inférieur au reste à financer. Avec l'offre 11, le montant financé dépasse le montant du projet de 600 €. Aussi des offres vont être annulées en totalité (elles passent dans l'état "Non retenue") ou partiellement (seule une partie de l'offre reste valide). Ce sont bien sûr les offres au taux le plus élevé qui sont supprimées, en commençant par la plus récente. Sur l'exemple, l'offre de rang 11 à 900 € va provoquer la suppression de 600 € sur les offres en cours. Il y a deux offres à 10% : rang 2 et 6. On commence donc par invalider l'offre de rang 6 (la plus récente). Elle passe dans

l'état "Non retenue". Il reste encore 300 € à invalider. Il seront pris sur l'offre de rang 2 qui reste dans l'état "En cours" mais avec seulement 700 € qui seront prêtés si aucune nouvelle offre n'était faite d'ici la fin des enchères.

Quand on fait une nouvelle offre, elle doit être acceptée en totalité. Sur l'exemple de la figure 2, si on choisit un taux de 10% pour l'offre de rang 11, son montant ne pourra pas être supérieur à 300 €. Si on veut faire une offre à 9,9%, on peut aller jusqu'à 1 600 €, les deux offres de taux 10% se trouvant annulées.

**2.6.1** Pour l'offre de rang 12, quel est le montant maximum que l'on peut proposer si le taux choisi est 4% ? Même question si le taux est 9,9% ?

**Solution :** Pour le taux à 4%, on peut faire une offre égale au montant du projet moins la somme des offres faites à 4%. On peut donc prêter  $10\,000 - 800 = 9\,200$ €.

Sur les taux à 10%, il reste seulement 700 € sur l'offre 5. On ne peut donc prêter que 700 € à 9,9%.

L'offre de rang 12 est faite à 7% avec un montant de 1500 euros. Indiquer l'état des différentes offres (NR pour non retenue ou EC pour en cours).

**Solution :** On annule les 700 € restant de l'offre 2. L'offre au taux juste inférieur est la 5 à 9,9%. On lui retire  $1500 - 700 = 800$  €. Il ne lui reste plus que 200 €.

En conclusion : toutes les offres sont en cours sauf 6 et 2. L'offre 5 est en cours mais avec un montant restant de 100 €.

**2.6.2** Donner la spécification du sous-programme qui correspond à faire une nouvelle offre sur un projet en précisant le taux et le montant du prêt.

On indiquera la précondition sur les paramètres de ce sous-programme pour que l'offre soit valide.

**Solution :**

```

1 Procédure ajouter_offre(projet: in out Projet; montant, taux: in Entier)
2     -- faire une nouvelle offre sur le projet au montant et taux précisés
3     --
4     -- Nécessite :
5     --     taux >= 40 && taux <= 1000
6     --     montant >= 20
7     --     montant Mod 1000 = 0
8     --     somme_offres_inferieures_taux(projet, taux) + montant <= projet.montant

```

**2.6.3** Donner les premiers niveaux de raffinement de ce sous-programme. On ne demande pas d'écrire le code mais bien d'identifier les grandes étapes et de les structurer avec la méthode des raffinages.

**Indication :** Pour gérer les invalidations provoquées par une nouvelle offre, il est pratique d'avoir les offres ordonnées dans l'ordre des taux décroissants et, pour un taux donné, de la plus récente à la plus ancienne.

**Solution :**

```

1 R1 : Comment « ajouter une offre »
2     Ajouter l'offre à la fin
3     Trier les offres dans l'ordre des taux décroissants, puis des rangs croissants
4     Mettre à jour l'état des offres
5     Trier les offres dans l'ordre croissant des rangs
6
7 R2 : Comment « Ajouter une offre à la fin »

```

```

8     initialiser_offre(projet.offres[projet.nb_offres], montant, taux);
9     projet.nb_offres <- projet.nb_offres + 1
10
11  R3 : Comment « Trier... »
12     -- on peut utiliser ici n'importe quel algorithme de tri, par exemple le
13     -- tri par sélection, le tri par insertion, etc. Nous choisissons le
14     -- tri à bulle.
15     Pour i <- 1 JusquÀ i = projet.nb_offres - 1 Faire
16         Pour j <- 1 JusquÀ j = projet.nb_offres - j Faire
17             Si offres j-1 plus grande que offres j Alors
18                 permuter les offres j-1 et j du projet
19             FinPour
20         FinPour
21     FinPour
22
23  R3 : Raffinage De « offre de rang1 plus grande que offre de rang 2 »
24     Si projet.offres[rang1].taux <> projet.offres[rang2].taux Alors
25         Résultat <- projet.offres[rang1].taux > projet.offres[rang2].taux Alors
26     Sinon -- Ce sont les rang qui permettent de décider
27         Résultat <- rang1 < rang2
28     FinSi
29
30  -- Remarque : le tri à bulle étant stable, il suffirait de ne comparer que les
31  -- taux. Un tri est stable si deux éléments égaux pour le critère de
32  -- comparaison conservent la même position relative dans le tableau avant et
33  -- après le tri.
34
35  R3 : Raffinage De « permuter les offres r1 et r2 d'un projet »
36     Variable
37         mémoire: Offre
38     Début
39         mémoire <- projet.offres[r1]
40         projet.offres[r1] <- projet.offres[r2]
41         projet.offres[r2] <- mémoire
42     Fin
43
44  R2 : Comment « Mettre à jour l'état des offres »
45     -- principe : parcourir les offres de la dernière à la première, tant
46     -- que le cumul des offres est inférieur strictement au montant, l'offre
47     -- est retenue.
48     -- que
49     Variable :
50         cumul: Entier
51     Début
52         cumul <- 0
53         Pour i <- projet.nb_offres - 1 JusquÀ i = 0 Faire
54             si cumul < projet.montant Alors
55                 projet.offres[i] <- EN_COURS
56             Sinon
57                 projet.offres[i] <- NON_RETENUE
58             FinSi
59             cumul <- cumul + projet.offres[i]
60         FinPour
61     Fin
62
63  R2 : Comment « Trier suivant le rang »
64     -- Même principe que l'autre tri mais en changeant le critère.
65
66  Remarque : Il est coûteux de trier deux fois le tableau. Pour éviter ceci, on
67  pourrait avoir un tableau d'entier. Les entiers correspondant au rang de
68  l'offre dans le premier tableau. Il suffit alors de trier ce tableau. On ne
69  modifie pas le tableau initial et on n'a pas besoin de faire le second tri.

```



70  
71 **De** plus la permutation est plus efficace puisqu'on copie des entiers et non des  
72 offres. Le critère est un peu plus coûteux puisqu'on a une indirection  
73 supplémentaire : l'élément du deuxième tableau donne le rang dans le premier  
74 tableau.