

Nom : _____

Groupe : _____

Examen 1 (20%)

Programmation en sciences

420-SN1-RE

Enseignant : Marie-Christine Bélanger

Département : Informatique

Hiver 2025

Consignes :

- Répondre sur le questionnaire pour la partie théorique.
- Un logiciel d'édition de code (Thonny) SANS assistance IA et la feuille de formules annotée à la main sont permis pour tout l'examen.
- Aucun accès à internet ni à d'autres fichiers que le document papier. Toute communication avec une personne physique, virtuelle ou une intelligence artificielle est interdite.
- Durée de l'examen : 2h, mais vous avez droit à 3h pour le faire.
- Les messages d'erreur du logiciel doivent être compris et résolus par l'étudiant. Toutefois, si un pépin qui sort du cadre du cours survenait, l'enseignante peut assister l'étudiant.
- Il se peut qu'il y ait trop de lignes ou pas assez pour répondre.
- Il n'est pas demandé de penser à gérer les cas d'exception non-spécifiés dans la question.
- L'optimisation de code n'est pas évaluée dans cet examen
- Les fautes de français de sont pas évaluées dans cet examen
- Le respect des conventions de noms pour les variables et pour les fonctions est souhaitable, mais non évalué
- Il y a 7 questions. Le total est sur 95 points.

SECTION DE QUESTIONS THÉORIQUES (35 POINTS)

Pour cette section, répondre sur la copie directement.

QUESTION 1. (8 points)

a) Observez l'exécution du code suivant :

```
1  
2 taille_cm = input("Donnez votre taille en centimètres: ")  
3 taille_metres = taille_cm/100
```

Console

```
>>> %Run -c $EDITOR_CONTENT  
Donnez votre taille en centimètres: 178  
Traceback (most recent call last):  
  File "<string>", line 3, in <module>  
TypeError: unsupported operand type(s) for /: 'str' and 'int'
```

Pourquoi le code des lignes 2 et 3 provoque cette erreur? Expliquer clairement la raison, en quelques mots, en français :

b) Observez le code suivant :

```
x = 2
def fonction_exemple(x):
    resultat = 10*x + 6
    return resultat

print(resultat)
```

En faisant rouler ce code, on obtient une erreur. Expliquer pourquoi et comment la corriger pour voir apparaître 26 en console ($2*10+6$)

QUESTION 2. (4 points)

Encerclez les noms de variables qui sont **valides** dans le code **ET** dans les conventions de code en Python vues en classe :

- a) Prix1
- b) prix_1
- c) 1er_prix
- d) prix_1ere_place
- e) prix-1
- f) prixPremierePlace

QUESTION 3. (8 points)

Vrai ou faux :

- a) Il est possible d'avoir une structure if sans qu'elle n'ait de else associé _____
- b) Lorsqu'on appelle une fonction à plusieurs paramètres, l'ordre dans lequel on place les valeurs de paramètres (arguments) dans les parenthèses est important _____
- c) Le code à l'intérieur d'une fonction ne s'exécute que si on l'appelle _____
- d) Dans un if, s'il y a des conditions entrecoupées de **and**, il suffit qu'une seule de ces conditions soit fausse pour que la condition complète soit fausse _____

Par exemple :

```
if .... and .... and ... :
```

- e) On peut appeler une fonction lorsqu'on est déjà dans une fonction _____
- f) Le symbole == sert à affecter une valeur dans une variable _____
- g) Dans une condition, c'est le symbole == qu'il faut mettre pour comparer deux valeurs _____

QUESTION 4. (15 points)

Voici le code d'une fonction qui applique un ajustement de prix à la caisse selon la Règle d'exactitude des prix. En effet, si le prix chargé à la caisse d'un magasin est différent du prix affiché, le client a droit à un remboursement de 10\$ si le produit coûte 10\$ ou plus ou bien l'item gratuit si l'article est moins de 10\$.

```
1
2
3
4
5 def remboursement(prix_article, prix_affichage):
6     if prix_affichage - prix_article > 0:
7         if prix_article < 10:
8             ajustement = prix_article
9         else:
10            ajustement = 10
11     else:
12         ajustement = 0
13     return ajustement
14
15
16
17
18
```

- a) **Directement dans l'image** ci-haut, **écrire la ligne de code** (à la main) qui teste la fonction avec un article dont le prix est de 6\$, mais dont le prix d'affichage est de 7\$. NE PAS poser la question à un utilisateur, faire un simple test d'appel de la fonction avec les valeurs données. Il faut écrire le code tel que le résultat du test s'afficherait dans la console.
- b) Mettre les numéros d'étapes séquentielles que votre programme va franchir dans l'ordre pour l'appel fait à la question a). Indiquez clairement les numéros à gauche des lignes.
- c) Quelle sera la valeur de retour (et donc la valeur affichée en console) après l'appel à la fonction fait en a)?
-
- d) Que retournerait la fonction si on l'appelait avec un prix d'affichage **identique** au prix de l'article?
-

SECTION DE QUESTIONS DE CODE (60 POINTS)

Dans cette section, l'important est de respecter les contraintes de chaque question. Ce qui n'est pas précisé explicitement est laissé à votre discrétion, aucune méthode ou structure imposée. Le nombre de variables nécessaires, le nombre d'étapes de calcul ou l'ordre dans lequel les conditions sont posées n'ont pas d'importance, tant que le code fait de quoi est demandé sans planter et que les tests fonctionnent. Vous n'êtes pas tenus de gérer les cas spéciaux non-spécifiés dans la question.

QUESTION 5. (22 points)

On souhaite déterminer la portée d'un projectile quand on connaît la vitesse de lancement et l'angle de départ. Pour se faire, suivre les étapes ci-dessous. Il est à noter que vous n'avez pas à vous soucier de la conversion des unités de mesure :

a) Écrire une fonction nommée `portee_projectile(...)` **DANS** laquelle on fait les étapes suivantes :

- Prend en paramètre une vitesse en m/s et un angle en degrés.
- Si l'angle fourni est supérieur ou égal à 90 ou s'il est inférieur à 0, **écrire le message** d'erreur (L'angle doit être entre 0 et 89.9) et **retourner** la valeur 0. Voir exemple 3.
- Sinon, on calcule et on **retourne** la portée du projectile comme ceci :

$$portee = \frac{v^2 \cdot \sin\left(2 \cdot angle \cdot \frac{\pi}{180}\right)}{g}$$

g est l'accélération gravitationnelle de 9.81 m/s². **Il est demandé de mettre la valeur de 9.81 dans une variable et d'utiliser la variable dans le calcul**

v est la vitesse en m/s

$angle$ est l'angle en degrés

$portee$ est la portée du projectile

*Note : la fonction sinus se code avec `math.sin()` de la librairie `math`. Idem pour π , `math.pi`

b) **À l'extérieur** de la fonction, poser 2 questions à l'utilisateur et placer les réponses dans 2 variables distinctes. Demander un angle et demander une vitesse initiale en m/s.

c) Passez vos 2 valeurs à la fonction et affichez le retour comme dans les exemples ci-dessous. Le retour (la portée) doit être arrondi à 2 décimales.

Exemple 1 :

```
Entrez l'angle initial en degrés: 45
Entrez la vitesse initiale en m/s: 10
La portée sera de 10.19 mètres
```

Exemple 2 :

```
Entrez l'angle initial en degrés: 38.2
Entrez la vitesse initiale en m/s: 40.5
La portée sera de 162.51 mètres
```

Exemple 3 :

```
Entrez l'angle initial en degrés: -50
Entrez la vitesse initiale en m/s: 15
L'angle doit être entre 0 et 89.9
La portée sera de 0.00 mètres
```

QUESTION 6. (22 points)

On souhaite déterminer la quantité d'un certain médicament qu'il faut donner à un animal domestique, chien ou chat.

- a) Faire une fonction nommée `medicament(...)` **DANS** laquelle on fait toutes les étapes suivantes :
- Prends en paramètre le type d'animal (chien ou chat) et son poids en kg.
 - Applique la logique suivante pour déterminer le nombre de doses à administrer:
 - o Si l'animal est un chien et qu'il pèse moins de 15 kg, on lui donnera 1 dose.
 - o Pour un chien de 15 à 45 kg, on donnera 2 doses.
 - o Pour un chat de moins de 4 kg, on donnera 0,5 dose.
 - o Pour un chat de 4 à 20 kg, on donnera 1 dose.
 - o Pour les autres cas, on ne donnera pas ce médicament (0 dose).
 - Il faudra **retourner** le nombre de doses du médicament selon les paramètres d'entrée.
- b) À l'extérieur de la fonction, tester la fonction et laisser les détails des tests pour TOUS les cas possibles. Les chiens de divers poids, les chats de divers poids et un autre type d'animal. **Ne pas poser la question à un utilisateur**, je veux voir votre succession d'appels de fonctions avec les valeurs choisies pour tout vérifier.

Note : il n'est pas demandé de prévoir un comportement particulier pour les mots écrits légèrement différemment, comme "Chien", "chiens", "CHAT" ou autres déclinaisons. Traitez seulement les mots : "chien" et "chat".

QUESTION 7. (16 points)

- a) Faire une fonction dont vous devez inventer le nom (nom significatif), **DANS** laquelle on fait toutes les étapes suivantes :
- Ne prend **rien** en entrée
 - Fait les tâches suivantes :
 - o Piger un premier nombre entier au hasard entre 1 et 10
 - o Demander à l'utilisateur de choisir un nombre entre entier 1 et 10. (il n'est pas demandé de gérer les utilisateurs qui trichent en mettant des nombres autres qu'entre 1 et 10)
 - o Si le nombre choisi par l'utilisateur est le même que celui pigé par l'ordinateur, afficher « Vous avez gagné! »
 - o Sinon, afficher « Vous avez perdu! Vous étiez à ... du chiffre pigé! ». Les trois petits points sont un calcul du nombre choisit moins le nombre pigé (voir exemple ci-bas)
 - **Aucun retour** demandé, que des affichages.
- b) Tester votre fonction en l'appelant. Un seul test demandé.

Exemples de déroulement quand on essaie le même test plusieurs fois (ne pas oublier que la distance au chiffre pigé est aléatoire) :

```
Console <
>>> %Run examen1-2025.py
Choisissez un nombre entre 1 et 10: 2
Vous avez perdu! Vous étiez à -2 du chiffre pigé!
```

```
Console <
>>> %Run examen1-2025.py
Choisissez un nombre entre 1 et 10: 9
Vous avez perdu! Vous étiez à 3 du chiffre pigé!
```

```
Console <
>>> %Run examen1-2025.py
Choisissez un nombre entre 1 et 10: 5
Vous avez gagné!
```