

NOM : Hanon Ymous
(000000)
Place : 0

#0000



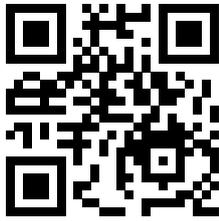
Programmation Orientée Objet (SMA/SPH) : SÉRIE NOTÉE

20 avril 2023

INSTRUCTIONS (à lire attentivement)

IMPORTANT! Veuillez suivre les instructions suivantes à la lettre sous peine de voir votre série annulée dans le cas contraire.

1. Vous disposez de 1h45 pour faire cette série notée (9h15 - 11h00).
2. Vous devez **écrire à l'encre noire ou bleu foncée**, pas de crayon ni d'autre couleur. N'utilisez **pas non plus de stylo effaçable** (perte de l'information à la chaleur).
3. Vous avez droit à toute documentation papier.
En revanche, vous ne pouvez pas utiliser d'ordinateur personnel, ni de téléphone portable, ni aucun autre matériel électronique.
4. Répondez aux questions directement sur la donnée; utilisez aussi le verso des feuilles, **MAIS** n'utilisez *que* le verso de la feuille sur laquelle se trouve la question, et non **pas** celui de la feuille précédente!
Ne joignez aucune feuille supplémentaire; **seul ce document sera corrigé**.
5. Lisez attentivement et *complètement* les questions de façon à ne faire que ce qui vous est demandé. Si l'énoncé ne vous paraît pas clair, ou si vous avez un doute, demandez des précisions à l'un(e) des assistant(e)s.
6. Cette série notée comporte deux exercices indépendants (pages 2 et 4), qui peuvent être traités dans n'importe quel ordre, mais qui ne rapportent pas la même chose (les points sont indiqués, le total est de 98). Les deux exercices comptent pour la note finale.



Exercice 1 – Solutions chimiques [sur 12 points]

Un chimiste souhaite qu'on l'aide à modéliser les mélanges de produits chimiques qu'il manipule. Le travail a été commencé, mais pas totalement fini (voir également la page ci-contre) :

```
#include <iostream>
#include <string>
#include <cmath>
using namespace std;

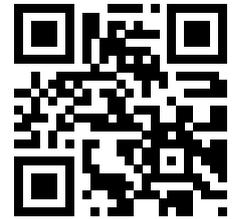
class Flacon
{
private:
    string nom;
    double volume;
    double pH;

public:
    Flacon(const string& unNom, double unVolume, double unPh)
        : nom(unNom), volume(unVolume), pH(unPh)
    {}

};

// ... SUITE SUR LA PAGE SUIVANTE ...
```

Ne pas écrire dans cette zone.



```
// ...vous pouvez encore écrire ici si nécessaire :

// =====
void afficher_melange(Flacon const& f1, Flacon const& f2)
{
    cout << "Si je mélange " << endl;
    cout << "    \" << f1 << \"\" << endl;
    cout << "avec" << endl;
    cout << "    \" << f2 << \"\" << endl;
    cout << "j'obtiens :" << endl;
    cout << "    \" << (f1 + f2) << \"\" << endl;
}
// =====
int main()
{
    Flacon flacon1("Eau",          600.0, 7.0);
    Flacon flacon2("Acide chlorhydrique", 500.0, 2.0);
    Flacon flacon3("Acide perchlorique", 800.0, 1.5);

    afficher_melange(flacon1, flacon2);
    afficher_melange(flacon2, flacon3);

    return 0;
}
```

Complétez le code précédent pour qu'il compile et qu'à l'exécution il produise :

```
Si je mélange
    "Eau : 600 ml, pH 7"
avec
    "Acide chlorhydrique : 500 ml, pH 2"
j'obtiens :
    "Eau + Acide chlorhydrique : 1100 ml, pH 2.34242"
Si je mélange
    "Acide chlorhydrique : 500 ml, pH 2"
avec
    "Acide perchlorique : 800 ml, pH 1.5"
j'obtiens :
    "Acide chlorhydrique + Acide perchlorique : 1300 ml, pH 1.63253"
```

La formule à utiliser pour le mélange (deux acides forts) est :

$$\text{pH} = -\log_{10} \left(\frac{\text{volume1} \times 10^{-\text{ph1}} + \text{volume2} \times 10^{-\text{ph2}}}{\text{volume1} + \text{volume2}} \right)$$

Note : la fonction \log_{10} s'écrit `log10()` en C++ et pour faire le calcul 10^{-x} utiliser l'expression `pow(10.0, -x)`.



Exercice 2 – Roulette [sur 86 points]

On s'intéresse ici à simuler des stratégies de jeu pour le jeu de la roulette. On vous demande pour cela d'écrire un programme en C++, en utilisant une approche « orientée objets » avec le moins de duplication de code possible.

Pour déjà donner une idée générale, voici un exemple de `main()` possible et le résultat attendu correspondant. Regardez le bien avant de commencer à coder et revenez y autant que nécessaire lors de la lecture de la suite de la donnée.

Pour que ce soit plus pratique, cet exemple est redonné en toute dernière dernière page.

```
int main() {
    Joueur M("Mortimer");
    // miser 100 jetons sur le 1
    Pleine p1(100, 1);
    M.set_strategie(p1);

    Joueur B("Blake");
    // miser 30 jetons sur les rouges
    Rouges p2(30);
    B.set_strategie(p2);

    Jeu j;
    j.participe(B);
    j.participe(M);

    for (auto tirage : { 12, 1, 31 }) {
        j.rien_ne_va_plus(tirage);
        j.affiche_resultats();
        cout << endl;
    }

    // miser 50 jetons sur le 17
    Pleine p3(50, 17);
    M.set_strategie(p3);
    j.rien_ne_va_plus(17);
    j.affiche_resultats();
    cout << endl;
}
```

```
Mortimer choisit la stratégie : mise 100 sur le 1
Blake choisit la stratégie : mise 30 sur les rouges
Croupier : le numéro tiré est le 12
Le joueur Blake mise 30 sur les rouges et gagne 30
Le joueur Mortimer mise 100 sur le 1 et perd sa mise

Croupier : le numéro tiré est le 1
Le joueur Blake mise 30 sur les rouges et gagne 30
Le joueur Mortimer mise 100 sur le 1 et gagne 3500

Croupier : le numéro tiré est le 31
Le joueur Blake mise 30 sur les rouges et perd sa mise
Le joueur Mortimer mise 100 sur le 1 et perd sa mise

Mortimer change de stratégie : mise 50 sur le 17
Croupier : le numéro tiré est le 17
Le joueur Blake mise 30 sur les rouges et perd sa mise
Le joueur Mortimer mise 50 sur le 17 et gagne 1750
```

La roulette est un jeu de hasard dans lequel chaque joueur mise¹ des jetons sur un ensemble de numéros de couleur rouge ou noire, compris entre 0 et 36 (sans couleur pour le 0).

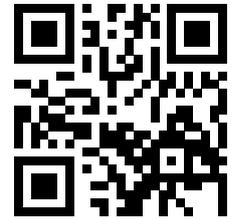
Quand les joueurs ont effectué leur mise, le croupier lance la bille et annonce « rien ne va plus »². À partir de ce moment, plus aucune mise n'est admise. Quand la bille s'arrête, les joueurs reçoivent leur gain³ ou perdent leur mise.

Les règles de la roulette déterminent les types de mises qui sont possibles. Chaque type de mises définit un sous-ensemble particulier de numéros, ainsi que le gain obtenu si le numéro tiré appartient à l'ensemble. Par exemple, une mise sur les rouges est gagnante si le numéro tiré est rouge ; elle rapporte alors une fois la somme mise.

1. “bet” in English.

2. “no more bets” in English.

3. “winnings” in English.



Un joueur qui effectue une mise gagnante remporte le gain correspondant et conserve la somme mise. Toutes les mises portant sur des ensembles de numéros ne contenant pas le numéro tiré sont perdues par les joueurs.

Question 2.1 – Les mises [sur 42 points]

Question 2.1.1 – Les mises générales [sur 17.5 points]

Une mise comprend un nombre de jetons misés, ainsi qu'un moyen de calculer le gain en fonction du numéro tiré et un moyen permettant d'afficher la mise⁴. Une fois fixé, le nombre de jetons misés ne peut être changé. Par ailleurs, ces deux fonctionnalités (calcul de gain et affichage) doivent pouvoir être spécifiques à chaque type de mises (voir plus loin).

Définissez une classe `Mise` correspondant la description ci-dessus. Cette classe est prévue pour être une description générale et ne devra pas pouvoir être instanciée. Elle devra par ailleurs avoir un moyen d'accéder au nombre de jetons misés et pouvoir être affichée au moyen de l'opérateur usuel `<<`.

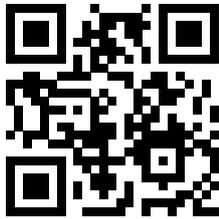
Par défaut, une `Mise` s'affiche simplement en affichant seulement « *mise* » suivi du nombre de jetons misés; p.ex. : `mise 100`.

Vous êtes libre de concevoir cette classe comme vous le souhaitez, tant qu'elle vérifie les contraintes ci-dessus et les principes de bonne programmation. Si un comportement ou une situation donnée n'est pas définie dans la consigne, vous êtes libre de choisir le comportement adéquat.

Cette remarque s'applique également à tout le reste de votre code.

Réponse : (vous pouvez répondre en deux colonnes si nécessaire, et avez aussi un peu de place au dos.)

4. Rappel: si nécessaire, voir l'exemple de déroulement fourni en début de donnée (et reproduit en dernière page).



Question 2.1.2 – Les mises spécifiques [sur 24.5 points]

Pour simplifier, nous allons supposer que seuls les deux types de mises suivantes sont possibles (en réalité, de nombreux autres types de mises sont possibles, mais nous ne les représenterons pas ici) :

- les mises portant sur un seul numéro ; ces mises rapportent 35 fois le nombre de jetons misés ;
- la mise sur les rouges, laquelle rapporte une fois le nombre de jetons misés ;
les numéros rouges sont 1, 3, 5, 7, 9, 12, 14, 16, 18, 19, 21, 23, 25, 27, 30, 32, 34 et 36 (dans votre code, vous pouvez raccourcir cette liste avec « ... »).

Créez (et définissez *complètement*) une classe `Pleine` pour représenter les mises sur un seul numéro et une classe `Rouges` pour représenter les mises sur les rouges. Les mises `Pleine` s'affichent au format suivant⁵ :

```
mise <nb jetons> sur le <numéro>
```

et les mises `Rouge` au format suivant :

```
mise <nb jetons> sur les rouges
```

Réponses :

5. Rappel: si nécessaire, voir l'exemple de déroulement fourni en début de donnée (et reproduit en dernière page).

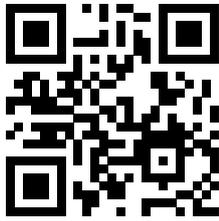
Question 2.1

Anonymisation : #0000
p. 7



Ne pas écrire dans cette zone.

suite au dos 



Question 2.2 – Les joueurs [sur 23 points]

On représentera les joueurs d'une partie de roulette par une classe `Joueur` qui possède un nom (non modifiable) et une stratégie de mise. Cette stratégie de mise utilisée par le joueur pourrait changer au moyen d'une méthode `set_stratégie()`⁶.

Lorsque `set_stratégie()` est appelée, un message sera affiché (sur `cout`) indiquant la mise choisie. Ce message commencera par le nom du joueur et « *choisit la stratégie* » si c'est la première fois ou « *change de stratégie* » s'il en change. Voir des exemples dans l'exemple de déroulement fourni en début de donnée.

Par ailleurs, la stratégie de mise, ainsi que le nom du joueur pourront être obtenus grâce à des « méthodes `get` ».

Indication : attention cependant aux « fuites d'encapsulation ».

Réponse :

6. Rappel: si nécessaire, voir l'exemple de déroulement fourni en début de donnée (et reproduit en dernière page).

Question 2.2

Anonymisation : #0000
p. 9



Ne pas écrire dans cette zone.

suite au dos 



Question 2.3 – Le jeu [sur 21 points]

Pour représenter une partie de roulette, créez une classe `Jeu` comprenant (au moins) :

- un ensemble de joueurs ;
- le dernier numéro tiré ;
- une méthode `participe()` pour y ajouter un joueur ;
- une méthode `rien_ne_va_plus()` qui fixe le numéro tiré ;
pour tester notre jeu, nous tricherons ici et la méthode `rien_ne_va_plus()` recevra en argument le numéro que l'on veut voir sortir et qui sera donc le numéro effectivement tiré⁷ ;
- une méthode `affiche_resultats()` qui affiche sur le terminal le déroulement d'un tirage comme dans l'exemple fourni en début de donnée.

Réponse :

Ne pas écrire dans cette zone.

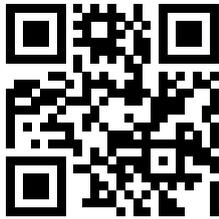
7. Rappel: si nécessaire, voir l'exemple de déroulement fourni en début de donnée (et reproduit en dernière page).

Question 2.3

Anonymisation : #0000
p. 11



Ne pas écrire dans cette zone.



```
int main() {
    Joueur M("Mortimer");
    // miser 100 jetons sur le 1
    Pleine p1(100, 1);
    M.set_strategie(p1);

    Joueur B("Blake");
    // miser 30 jetons sur les rouges
    Rouges p2(30);
    B.set_strategie(p2);

    Jeu j;
    j.participe(B);
    j.participe(M);

    for (auto tirage : { 12, 1, 31 }) {
        j.rien_ne_va_plus(tirage);
        j.affiche_resultats();
        cout << endl;
    }

    // miser 50 jetons sur le 17
    Pleine p3(50, 17);
    M.set_strategie(p3);
    j.rien_ne_va_plus(17);
    j.affiche_resultats();
    cout << endl;
}
```

Mortimer choisit la stratégie : mise 100 sur le 1
Blake choisit la stratégie : mise 30 sur les rouges
Croupier : le numéro tiré est le 12
Le joueur Blake mise 30 sur les rouges et gagne 30
Le joueur Mortimer mise 100 sur le 1 et perd sa mise

Croupier : le numéro tiré est le 1
Le joueur Blake mise 30 sur les rouges et gagne 30
Le joueur Mortimer mise 100 sur le 1 et gagne 3500

Croupier : le numéro tiré est le 31
Le joueur Blake mise 30 sur les rouges et perd sa mise
Le joueur Mortimer mise 100 sur le 1 et perd sa mise

Mortimer change de stratégie : mise 50 sur le 17
Croupier : le numéro tiré est le 17
Le joueur Blake mise 30 sur les rouges et perd sa mise
Le joueur Mortimer mise 50 sur le 17 et gagne 1750

Ne pas écrire dans cette zone.