

Examen M1 / C++

La propreté de la programmation sera pris en compte dans l'évaluation.

Exercice 1 Quels sont les paradigmes supportés par C++ ?

Exercice 2 Donnez le code de la fonction `facto` transformant un nombre n en son factoriel $n! = n \times (n-1) \times \dots \times 2$, par exemple on déclare `int x=4`; puis après l'appel on a $x = 24$, dans les cas suivants:

- 1) L'appel avait été effectué par l'instruction `facto(&x)`;
- 2) L'appel avait été effectué par l'instruction `facto(x)`;
- 3) L'appel avait été effectué par l'instruction `x=facto(x)`;

Exercice 3 Faire une fonction `arrangement(int m)` affichant toutes les valeurs de

$$ar(n,p) := \binom{n}{p} \times p! = n \times \dots \times (n-p+1)$$

pour $n = 0, 1, \dots, m$ et $0 \leq p \leq n$. Par exemple l'appel `arrangement(4)` affiche :

```
1
1 1
1 2 2
1 3 6 6
1 4 12 24 24
```

On procédera impérativement de la façon décrite ci-dessous.

- 1) On utilisera un tableau de tableaux via un pointeur de pointeurs `int** mat`. Le tableau de tableaux contient $m+1$ tableaux `mat[i]` de taille $i+1$ pour $i = 0, 1, \dots, m$. On fera l'allocation de mémoire avec $m+2$ appels à `new` et la désallocation avec $m+2$ appels à `delete[]`. Par exemple `int** mat = new int* [m+1]`; pour le tableau de tableaux...
- 2) On initialisera `mat` de la façon suivante : `mat[i][0]=1` pour tout i . Puis on complètera `mat` avec la relation suivante $ar(n,p) = p \times ar(n-1,p-1) + ar(n-1,p)$ avec $ar(n-1,p) = 0$ pour $p = n$. Ainsi la valeur de `mat[i][j]` sera égale à $ar(i,j)$.
- 3) Faire l'affichage.

Exercice 4 1) Implémenter une classe `Rationnels` modélisant l'ensemble $\mathbb{Q} = \{p/q : p \in \mathbb{Z}, q \in \mathbb{N} \setminus \{0\}\}$ ayant comme attributs privés `int p,q`; et ayant deux opérateurs membres `+=` et `*=` modélisant l'addition et la multiplication dans \mathbb{Q} . Par exemple:

```
Rationnels a(2,3), b(1,5), c(3,1), d(12,15);
a*=b; // a=2/15
```

2) Implémenter des opérateurs binaires `+` et `*` modélisant l'addition et la multiplication dans \mathbb{Q} mais n'ayant pas accès aux attributs privés de `Rationnels`. Les opérateurs binaires devront supporter le chaînage, par exemple on doit pouvoir faire:

```
a= a+b+c; // a=28/15
```

3) Faire une fonction membre `reduct` qui réduit la fraction. Par exemple

```
d.reduct(); // d=4/5
```