Nom : _____ (_____/5)

STR – Q03

Voici deux possibilités pour implémenter `new` et `delete` de manière globale[1]. Choisissez **soit** l'option `A`, **soit** l'option `B` ci-dessous, **pas les deux!** (*indiquez clairement laquelle!*), et implémentez ces deux opérateurs (ceci vise celles et ceux qui sont confortables avec la syntaxe du langage).

Option `A` : la mémoire nouvellement allouée avec `new` doit être pleine de `0x00` et la mémoire libérée avec `delete` doit être remplie `0xff` avant d'être libérée.

Option `B` : la mémoire nouvellement allouée avec `new` doit être précédée et suivie d'un bloc de `32` bits contenant la valeur `0xdeadc0de`; `delete` doit agir en conséquence.

Consignes :

- Réalisez l'allocation brute avec `std::malloc()` et la libération correspondante avec `std::free()`.
- Par souci de simplicité, tenez ce qui suit pour acquis :
  - Les `#include` et les `using` sont déjà faits.
  - Huit bits par *byte*, donc `std::numeric_limits<unsigned char>::digits==8`
  - `sizeof(int)==4`
  - La mémoire retournée par `std::malloc()` est alignée sur une frontière convenant au pire cas possible, soit `std::max_align_t`.

[1] Les versions `new[]` et `delete[]` ne sont pas exigées.