

Devoir 1:

Exercice :

Le but de ce devoir est de déterminer de manière pratique combien de paramètres une fonction peut accepter avant de provoquer une erreur ou un comportement inattendu. Vous devez écrire une fonction qui accepte plusieurs paramètres, puis augmenter progressivement leur nombre jusqu'à ce qu'une limite soit atteinte. Observez et documentez à quel moment une erreur (préciser le type d'erreur trouvé) se produit ou si le programme devient instable. *Expliquez les résultats obtenus en fonction des limitations liées à l'architecture de la machine et à la gestion de la mémoire.*

Objectif :

Ce projet vise à déterminer combien d'éléments une fonction peut accepter avant de provoquer une erreur ou un comportement inattendu. Nous avons utilisé la mémoire dynamique avec un tableau (`vector<int>`) pour observer comment le programme réagit à des allocations de mémoire de plus en plus importantes.

Code Source :

Voici le code utilisé pour réaliser ce test :

```
✓ #include <iostream>
  #include <stdexcept>
  #include <vector>
  using namespace std;

✓ void afficherTaille(const vector<int>& tableau) {
  |   cout << "Fonction appelee avec " << tableau.size() << " elements." << endl;
  | }
✓ int main() {
  | try {
  |     // Boucle qui crée des tableaux de plus en plus grands
  |     for (size_t taille = 1;; taille *= 3) {
  |         vector<int> tableau(taille, 1);
  |         afficherTaille(tableau);
  |     }
  | } catch (const bad_alloc& erreur) {
  |     cout << "Erreur d'allocation de memoire : " << erreur.what() << endl;
  | }

  | return 0;
  | }
}
```

Résultats:

Le programme a pu accepter jusqu'à environ 10 milliards d'éléments avant de provoquer une erreur d'allocation de mémoire. L'erreur rencontrée est de type `bad_alloc`, ce qui signifie que la mémoire disponible pour l'allocation était insuffisante.

```
Fonction called with 1 parametres.
Fonction called with 3 parametres.
Fonction called with 9 parametres.
Fonction called with 27 parametres.
Fonction called with 81 parametres.
Fonction called with 243 parametres.
Fonction called with 729 parametres.
Fonction called with 2187 parametres.
Fonction called with 6561 parametres.
Fonction called with 19683 parametres.
Fonction called with 59049 parametres.
Fonction called with 177147 parametres.
Fonction called with 531441 parametres.
Fonction called with 1594323 parametres.
Fonction called with 4782969 parametres.
Fonction called with 14348907 parametres.
Fonction called with 43046721 parametres.
Fonction called with 129140163 parametres.
Fonction called with 387420489 parametres.
Fonction called with 1162261467 parametres.
Fonction called with 3486784401 parametres.
Fonction called with 10460353203 parametres.
Memory allocation error: std::bad_alloc
```

Analyse des résultats :

L'erreur bad_alloc se produit parce que le système ne peut plus allouer de mémoire pour le tableau sur le tas . Cette limitation est liée à la quantité de RAM disponible sur la machine, ainsi qu'à la gestion de la mémoire par le système d'exploitation.

Conclusion :

Ce projet démontre l'importance de comprendre comment la mémoire est gérée dans un programme C++. Les erreurs d'allocation de mémoire peuvent survenir si nous ne sommes pas attentifs à la taille des données que nous essayons de manipuler. Cela souligne également la nécessité d'une gestion appropriée de la mémoire dynamique pour éviter les fuites et les comportements imprévus.