

Com base no programa-exemplo da questão anterior, proponha uma aplicação que permita comparar o desempenho das funções tradicionais de alocação/liberação de memória da linguagem c (malloc/free) com as funções de gerenciamento de memória da biblioteca que você escolheu. A aplicação deverá ser capaz de ressaltar possíveis atrasos de tempo provenientes do coletor de lixo utilizado.

### Programa (Input):

```
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>
#include <gc/gc.h>
#include <time.h>

int main(void)
{
    int i;
    const int size = 10000000;
    clock_t t1, t2;

    t1 = clock();
    GC_INIT();
    for (i = 0; i < size; ++i){
        int **p = GC_MALLOC(sizeof(int));
    }
    t1 = clock() - t1;
    printf("%f ms", ((float)t1)/(CLOCKS_PER_SEC/1000));

    t2 = clock();
    for (i = 0; i < size; ++i){
        int **q = malloc(sizeof(int));
        free(q);
    }
    t2 = clock() - t2;
    printf("%f ms", ((float)t2)/(CLOCKS_PER_SEC/1000));

    return 0;
}
```

Com a implementação da biblioteca time.h, o uso da função clock antes e depois de cada função de alocação de memória (A

primeira usando as funções `GC_INIT` e `GC_MALLOC` da biblioteca `libGC` e a segunda usando as funções `malloc/free`) permite comparar o tempo de execução de cada uma.