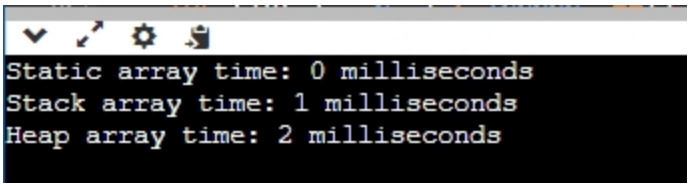
HW4

Write three functions in C or C++: one that declares a large array statically, one that declares the same large array on the stack, and one that creates the same large array from the heap. Call each of the subprograms a large number of times (at least 100,000) and output the time required by each. Explain the results.

C++

```cpp
#include <iostream>
#include <chrono>

using namespace std;
using namespace std::chrono;

void staticArray() {
    static int arr[1024 * 10];
}

void stackArray() {
    int arr[1024 * 10];
}

void heapArray() {
    int* arr = new int[1024 * 10];
    delete[] arr;
}

int main() {
    auto start = high_resolution_clock::now();
    for (int i = 0; i < 100000; ++i) {
        staticArray();
    }
    auto stop = high_resolution_clock::now();
    auto duration = duration_cast<milliseconds>(stop - start);
    cout << "Static array time: " << duration.count() << " milliseconds" << endl;

    start = high_resolution_clock::now();
    for (int i = 0; i < 100000; ++i) {
        stackArray();
    }
    stop = high_resolution_clock::now();
    duration = duration_cast<milliseconds>(stop - start);
    cout << "Stack array time: " << duration.count() << " milliseconds" << endl;

    start = high_resolution_clock::now();
    for (int i = 0; i < 100000; ++i) {
        heapArray();
    }
    stop = high_resolution_clock::now();
    duration = duration_cast<milliseconds>(stop - start);
    cout << "Heap array time: " << duration.count() << " milliseconds" << endl;

    return 0;
}

```

output



Explain the results

Static : การประกาศอาร์เรย์แบบ static ในฟังก์ชัน staticArray() ที่บรรทัดที่ 7 นั้นมีความไวที่สุด เนื่องจาก การจัดสรรหน่วยความจำเกิดขึ้นในขั้นตอนของการโหลดโปรแกรม ซึ่งหมายความว่า อาร์เรย์จะถูกสร้างขึ้นและจัดเก็บในหน่วยความจำที่ถูกกำหนดไว้ตั้งแต่เริ่มต้นของโปรแกรม ซึ่งทำให้ไม่มีความล่าช้าในการจัดสรรหน่วยความจำเมื่อฟังก์ชันถูกเรียกใช้ สิ่งนี้ช่วยลดการใช้งานทรัพยากรและเพิ่มประสิทธิภาพในระหว่างการดำเนินการ

Stack : การประกาศอาร์เรย์แบบ stack ในฟังก์ชั่น stackArray() ที่บรรทัดที่ 11 นั้นมีความไวรองลงมาเนื่องจากการสร้าง array แบบ stack นั้นถูกสร้างขึ้นมาในระหว่าง run time และเป็นการจัดการหน่วยความจำแบบ Dynamic

Stack : การประกาศอาร์เรย์แบบ heap ในฟังก์ชั่น heapArray() ที่บรรทัดที่ 15 นั้นมีความไวต่ำที่สุดเนื่องจากการสร้าง array แบบ heap นั้นมีความซับซ้อนยิ่งกว่า โดยการจัดสรรของระบบในช่วง runtime