Execution Instructions

Run The CODE in any C++ compiler like this-> (https://www.programiz.com/cpp-programming/online-compiler/)

1. Once the compilation is successful, run the executable code.

2. Then the program will prompt you to enter the number of points and the coordinates for each point. Follow the prompts as instructed.

3. After entering the points, the program will print the top-right point and display the elapsed time for different values of 'n'.

4. Examine the output to see the top-right point and the time elapsed for various values of 'n'.

The screenshots below demonstrate the execution outcomes

main.cpp

Run

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

using namespace std;

class Point {
public:
    int x, y;

    Point() : x(0), y(0) {}

    Point(int x, int y) : x(x), y(y) {}
};

class DynamicArray {
public:
    DynamicArray(int capacity = 10) {
        array_ = vector<Point>(capacity);
        size_ = 0;
    }

    void append(Point point) {
        if (size_ >= array_.size()) {
            array_.resize(array_.size() * 2);
        }
        array_[size_] = point;
        size_++;
    }

    Point get(int index) {
        if (index < 0 || index >= size_) {
```

Output

Clear

```
/tmp/VDGauVFr0U.o
Enter the number of points: 4
Enter point 1: -3 -3
Enter point 2: -1 1
Enter point 3: 2 -2
Enter point 4: 0 0
Top-right point is: (2, -2)
Time elapsed for n = 10: 77310 nanoseconds
Time elapsed for n = 100: 1256350 nanoseconds
Time elapsed for n = 1000: 31840348 nanoseconds
Time elapsed for n = 10000: 452543897 nanoseconds
Time elapsed for n = 100000: 17838570427 nanoseconds
```

main.cpp

Run

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

using namespace std;

class Point {
public:
    int x, y;

    Point() : x(0), y(0) {}

    Point(int x, int y) : x(x), y(y) {}
};

class DynamicArray {
public:
    DynamicArray(int capacity = 10) {
        array_ = vector<Point>(capacity);
        size_ = 0;
    }

    void append(Point point) {
        if (size_ >= array_.size()) {
            array_.resize(array_.size() * 2);
        }
        array_[size_] = point;
        size_++;
    }

    Point get(int index) {
        if (index < 0 || index >= size_) {
```

Output

Clear

```
/tmp/VDGauVFr0U.o
Enter the number of points: 5
Enter point 1: 0 0
Enter point 2: 1 1
Enter point 3: 2 2
Enter point 4: -1 -1
Enter point 5: -2 -2
Top-right point is: (2, 2)
Time elapsed for n = 10: 66470 nanoseconds
Time elapsed for n = 100: 1190870 nanoseconds
Time elapsed for n = 1000: 24748358 nanoseconds
Time elapsed for n = 10000: 592029665 nanoseconds
Time elapsed for n = 100000: 22386800964 nanoseconds
```

main.cpp

Run

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

using namespace std;

class Point {
public:
    int x, y;

    Point() : x(0), y(0) {}

    Point(int x, int y) : x(x), y(y) {}
};

class DynamicArray {
public:
    DynamicArray(int capacity = 10) {
        array_ = vector<Point>(capacity);
        size_ = 0;
    }

    void append(Point point) {
        if (size_ >= array_.size()) {
            array_.resize(array_.size() * 2);
        }
        array_[size_] = point;
        size_++;
    }

    Point get(int index) {
        if (index < 0 || index >= size_) {
```

Output

Clear

```
/tmp/VDGauVFr0U.o
Enter the number of points: 3
Enter point 1: -5 -10
Enter point 2: 3 7
Enter point 3: -1 5
Top-right point is: (3, 7)
Time elapsed for n = 10: 51589 nanoseconds
Time elapsed for n = 100: 786840 nanoseconds
Time elapsed for n = 1000: 14714629 nanoseconds
Time elapsed for n = 10000: 438960357 nanoseconds
Time elapsed for n = 100000: 15145083633 nanoseconds
```