Question-1 => Implement a dynamic array (that is a C++ vector). You are only allowed to use C style arrays. Assume the datatype is an int.

```
#include <iostream>

class DynamicArray {
private:
   int* array;
   int capacity;
   int size;

public:
   DynamicArray() : capacity(1), size(0) {
      array = new int[capacity];
   }

   ~DynamicArray() {
      delete[] array;
   }

   void push_back(int element) {
      if (size == capacity) {
         // Double the capacity
         int new_capacity = capacity * 2;
         int* new_array = new int[new_capacity];
         for (int i = 0; i < size; ++i) {
            new_array[i] = array[i];
         }
         delete[] array;
         array = new_array;
         capacity = new_capacity;
      }
      array[size++] = element;
   }

   int& operator[](int index) {
      if (index < 0 || index >= size) {
         std::cerr << "Index out of bounds\n";
         exit(1);
      }
```

```
        return array[index];
    }

    int getSize() const {
        return size;
    }
};

int main() {
    DynamicArray dynArray;

    // Adding elements
    for (int i = 0; i < 10; ++i) {
        dynArray.push_back(i);
    }

    // Accessing elements
    for (int i = 0; i < dynArray.getSize(); ++i) {
        std::cout << dynArray[i] << " ";
    }
    std::cout << std::endl;

    return 0;
}
```