

Find the Output of the C++ code segments. Assume that all things needed are included.

1	<pre> int main() { vector<int> vec; for (int i = 0; i < 5; i++) { vec.push_back(i); } vector<int>::iterator a; for (a = vec.begin(); a != vec.end(); a++) { *a++; } for (a = vec.end(); a != vec.begin(); a--) { cout << *a << " "; } cout << endl; } </pre>	
2	<pre> int main() { int *a = new int [10]; for (int i = 9; i >= 0; --i) { a[i] = i * 2; } for (int i = 0; i < 10; i++) { cout << ++*a << endl; } } </pre>	
3	<pre> int main() { int *a = new int[10]; int b = {5,4,3,2,1}; int *c = &b[1]; for (int i = 0; i < 10; i++) { *(a++) = i; } c++; a[2] += c; for (int i = 0; i < 10; i++) { cout << a[i] << " "; } cout << endl; } </pre>	
4	<pre> int main() { int a = 0; for (int i = a; i < 10; i++) { for (int j = a + 1; j < 11; j++) { a++; } } cout << a << endl; } </pre>	
5	<pre> class A { int z; public: A(int a): z(a) {} } </pre>	

	<pre> &A operator+(const &A ref) { this->z += ref.z; return *this; } ~A() { cout << z << endl; } } foo(12); int main() { A b(23); b + foo; } </pre>	
6	<pre> class A { public: A() {cout << "Constructed" << endl;} ~A() {cout << "Destructed" << endl;} } int main() { A a, *b; for (int i = 0; i < 1; i++) { A c; } } </pre>	
7	<pre> class A { protected: int z; public: A() : z(1) {} virtual void run() {} void peek() {cout << z << endl;} }; class B : public A { int y; public: B(int a) : y(a - 1), z(a) {} void run() {cout << y << " " << z << endl;} } int main() { A a, *c; B b(14); c = static_cast<A*>(&b); c->peek(); b.run(); } </pre>	
8	<pre> char s = "Hello"; int main() { int s = 500; cout << s++ << endl; } </pre>	
9	<pre> class A { </pre>	

	<pre> int a,b; public: A(int z, int x) : a(z), b(x) { if (x == 0) { throw "X is Zero"; } } } int main() { int a = 500; int b = 0; try { A c(a, b); } catch (char* e) { cout << e << endl; } } </pre>	
--	---	--

Find the Errors in the C++ Code Segments

1	<pre> int a; a += 4; cout << a << endl; </pre>	
2	<pre> int *a = {0,1,2,3,4,5}; for (int i = 0; i < 6; i++) { *(++a) = 2; } cout << a[2] << endl; </pre>	
3	<pre> const int foo = 9000; foo++; </pre>	
4	<pre> int *a = new int[3]; for (int i = 0; i < 3; i++) { a[i] = i - 1; } delete a; for (int i = 0; i << 3; i++) { cout << a << " "; } </pre>	
5	<pre> class A{ int a = 0; public: A() {} } class B { int b = 1; public: A() {} } int main() { A *a; B b; a = &b; } </pre>	
6	<pre> class B { </pre>	

	<pre> int swe = 234; public: B(int a) : swe(a); } </pre>	
7	<pre> class A { int f; A() {} ~A() {cout << "Destructed" << endl;} } </pre>	
8	<pre> int a = 5; cout << ++a << " " << a++ << endl; </pre>	
9	<pre> void foo(int a) const { a++; for (int i = 0; i < a; i++) { cout << "240"; } } </pre>	