

**100%** Money Back  
**Guarantee**

**Vendor:** C++ Institute

**Exam Code:** CPP

**Exam Name:** C++ Certified Professional Programmer

**Version:** Demo

**Topic 1, Volume A****QUESTION NO: 1**

What happens when you attempt to compile and run the following code?

```
#include <iostream>

#include <set>

#include <vector>

using namespace std;

int main(){

int t[] ={ 3, 4, 2, 1, 6, 5, 7, 9, 8, 0 };

vector<int>v(t, t+10);

multiset<int> s1(v.begin(),v.end());

s1.insert(v.begin(),v.end());

pair<multiset<int>::iterator,multiset<int>::iterator> range;

range = s1.equal_range(6);

while (range.first != range.second) {

cout<<*range.first<<" "; range.first++;

}

return 0;

}
```

- A. program outputs: 6 6
- B. program outputs: 5 7
- C. program outputs: 5 5 6 6 7 7
- D. program outputs: 5 5 7 7
- E. program outputs: 1 1 6 6 5 5

**Answer: A**

**QUESTION NO: 2**

What happens when you attempt to compile and run the following code?

```
#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

template<class T>struct Out {
    ostream & out;
    Out(ostream & o): out(o){}
    void operator()(const T & val ) {
        out<<val<<" ";
    }
};

struct Sequence {
    int start;
    Sequence(int start):start(start){}
    int operator()() {
        return start++ ; };
};

int main() {
    vector<int> v1(10);
    generate(v1.rbegin(), v1.rend(), Sequence(1));
    rotate(v1.begin(),v1.begin() + 1, v1.end() );
    for_each(v1.begin(), v1.end(), Out<int>(cout) );cout<<endl;
    return 0;
}
```

Program outputs:

- A. 1 2 3 4 5 6 7 8 9 10
- B. 10 9 8 7 6 5 4 3 2 1
- C. 9 8 7 6 5 4 3 2 1 10
- D. 1 10 9 8 7 6 5 4 3 2

**Answer: C**

### QUESTION NO: 3

What happens when you attempt to compile and run the following code?

```
#include <iostream>

#include <fstream>

#include <string>

#include <list>

#include <algorithm>

#include <iomanip>

using namespace std;

class B { int val;

public:

B(int v=0):val(v){}

int getV() const {return val;}

operator int() const { return val; }};

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) {out<<setw(3)<<hex<<val; } };
```

```
int main () {  
  
int t[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};  
  
fstream f("test.out", ios::trunc|ios::out);  
  
list<B> l(t, t+10);  
  
for_each(l.begin(), l.end(), Out<B>(f));  
  
f.close();  
  
f.open("test.out");  
  
for( ; f.good() ; ) {  
  
    B i;  
  
    f>>i;  
  
    cout<<i<<" ";  
  
}  
  
f.close();  
  
return 0;  
  
}
```

- A. file test.out will be opened writing
- B. file test.out will be truncated
- C. file test.out will be opened for reading
- D. compilation error
- E. program will display sequence 1 2 3 4 5 6 7 8 9 10

**Answer: D**

#### QUESTION NO: 4

What will happen when you attempt to compile and run the code below, assuming that you enter the following sequence: one two three<enter>?

```
#include <iostream>
```

```
#include <string>

using namespace std;

int main ()
{
string a;

cin>>a;

cout<<a<<endl;

return 0;
}
```

Program will output:

- A. one
- B. one two three
- C. runtime exception
- D. compilation error
- E. the result is unspecified

**Answer: A**

### QUESTION NO: 5

What will happen when you attempt to compile and run the following code?

```
#include <iostream>

#include <map>

#include <vector>

#include <sstream>

#include <string>

using namespace std;
```

```
int main() {  
  
int t[] = { 3, 4, 2, 1, 0, 3, 4, 1, 2, 0 };  
  
vector<int> v(t, t + 10);  
  
multimap<int, string> m;  
  
for (vector<int>::iterator i = v.begin(); i != v.end(); i++) {  
  
stringstream s;s << *i << *i;  
  
m.insert(pair<int, string>(*i, s.str()));  
  
}  
  
pair<multimap<int, string>::iterator, multimap<int, string>::iterator> range;  
  
range = m.equal_range(2);  
  
for (multimap<int, string>::iterator i = range.first; i != range.second; i++) {  
  
cout << i->first << " ";  
  
}  
  
return 0;  
  
}
```

The output will be:

- A. 2 2
- B. 1 2
- C. 1 3
- D. 2
- E. 0 2

**Answer: A**

#### QUESTION NO: 6

What happens when you attempt to compile and run the following code?

```
#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

class B { int val;

public:

B(int v):val(v){}

int getV() const {return val;} bool operator < (const B & v) const { return val>v.val;} };

ostream & operator <<(ostream & out, const B & v) { out<<v.getV(); return out;}

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) { out<<val<<" "; };

};

int main() {

B t1[]={3,2,4,1,5};

B t2[]={5,6,8,2,1};

vector<B> v1(10,0);

sort(t1, t1+5);

sort(t2, t2+5);

set_intersection(t1,t1+5,t2,t2+5,v1.begin());

for_each(v1.begin(), v1.end(), Out<B>(cout));cout<<endl;

return 0;

}
```

Program outputs:

**A.** compilation error



- B. 1 2 3 4 5 6 8 0 0 0
- C. 1 2 3 4 5 6 8 2 1 0
- D. 5 2 1 0 0 0 0 0 0
- E. 1 2 5 0 0 0 0 0 0

**Answer: D**

### QUESTION NO: 7

What happens when you attempt to compile and run the following code?

```
#include <list>

#include <vector>

#include <iostream>

using namespace std;

int main ()

{

int t[] = {1, 2 ,3 ,4 ,5};

vector<int>v1(t, t+5);

list<int>l1;

l1.assign(v1.end(), v1.begin());

for(int i=0; i<l1.size(); i++)

{

cout<<l1.at(i)<<" ";

}

cout<<endl;

return 0;

}
```

- A. program displays 5 4 3 2 1
- B. program displays 1 2 3 4 5
- C. compilation error
- D. segmentation fault runtime exception

**Answer: C**

### QUESTION NO: 8

What happens when you attempt to compile and run the following code?

```
#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

class B { int val;

public:

B(int v):val(v){}

int getV() const {return val;} bool operator < (const B & v) const { return val<v.val;} };

ostream & operator <<(ostream & out, const B & v) { out<<v.getV(); return out;}

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) { out<<val<<" "; } };

int main() {

B t1[]={3,2,4,1,5};

B t2[]={6,10,8,7,9};
```

```
vector<B> v1(10);  
  
sort(t1, t1+5);  
  
sort(t2, t2+5);  
  
merge(t1,t1+5,t2,t2+5,v1.begin());  
  
for_each(v1.begin(), v1.end(), Out<B>(cout));cout<<endl;  
  
return 0;  
  
}
```

Program outputs:

- A. 1 2 3 4 5 6 10 8 7 9
- B. 3 2 4 1 5 6 7 8 9 10
- C. 3 2 4 1 5 6 10 8 7 9
- D. 1 2 3 4 5 6 7 8 9 10
- E. compilation error

**Answer: E**

### QUESTION NO: 9

Which sentence is correct about the code below?

```
#include <iostream>  
  
#include <algorithm>  
  
#include <vector>  
  
using namespace std;  
  
class A {  
  
int a;  
  
public:  
  
A(int a) : a(a) {}  
  
int getA() const { return a; }
```

```
void setA(int a) { this->a = a; }

/* Insert Code Here */

};

struct add10 { void operator()(A & a) { a.setA(a.getA() + 10); } };

int main() {

int t[] = { 10, 5, 9, 6, 2, 4, 7, 8, 3, 1 };

vector<A> v1(t, t + 10);

for_each(v1.begin(), v1.end(), add10());

vector<A>::iterator it = find(v1.begin(), v1.end(), A(7));

cout << it->getA() << endl;

return 0;

}
```

- A. it will compile and print 7
- B. it will not compile
- C. it will compile but the program result is unpredictable
- D. adding code:

```
bool operator !=(const A & b) const {
if (this->a != b.a) { return true; } return false; }
at Place 1 will allow the program to compile
```

**Answer: B**

### QUESTION NO: 10

What happens when you attempt to compile and run the following code?

```
#include <iostream>

#include <algorithm>
```

```
#include <vector>

using namespace std;

void myfunction(int i) {

cout << " " << i;

}

void multiply (int a) {

a*2;

}

int main() {

int t[] = { 10, 5, 9, 6, 2, 4, 7, 8, 3, 1 };

vector<int> v1(t, t+10);

for_each(v1.begin(), v1.end(), multiply);

iter_swap(v1.begin(),t+9);

for_each(v1.begin(), v1.end(), myfunction);

return 0;

}
```

Program outputs:

- A. 1 5 9 6 2 4 7 8 3 1
- B. compilation error
- C. 1 2 3 4 5 6 7 8 9 10
- D. 10 9 8 7 6 5 4 3 2 1
- E. 10 5 9 6 2 4 7 8 3 1

**Answer: A**

**QUESTION NO: 11**

What happens when you attempt to compile and run the following code?

```
#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) { out<<val<<" "; } };

int main() {

int t[]={3,2,4,1,5,10,9,7,8,6};

vector<int> v1(t,t+10);

cout<<*max_element(v1.begin(), v1.end());

return 0;

}
```

Program outputs:

- A. 3
- B. 1
- C. 6
- D. 10
- E. compilation error

**Answer: D**

**QUESTION NO: 12**

What happens when you attempt to compile and run the following code?

```
#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) { out<<val<<" "; } };

int main() {

int t1[]={3,2,4,1,5};

int t2[]={5,6,8,2,1};

vector<int> v1(10);

sort(t1, t1+5);

sort(t2, t2+5);

set_intersection(t1,t1+5,t2,t2+5,v1.begin());

for_each(v1.begin(), v1.end(), Out<int>(cout));cout<<endl;

return 0;

}
```

Program outputs:

- A. compilation error
- B. 1 2 3 4 5 6 8 0 0 0
- C. 1 2 3 4 5 6 8 2 1 0
- D. 1 1 2 2 3 4 5 5 6 8
- E. 1 2 5 0 0 0 0 0 0 0

**Answer: E**

**QUESTION NO: 13**

What happens when you attempt to compile and run the following code?

```
#include <iostream>

#include <algorithm>

#include <vector>

#include <deque>

#include <set>

using namespace std;

void myfunction(int i) {

cout << " " << i;

}

int main() {

int t[] = { 10, 5, 9, 6, 2, 4, 7, 8, 3, 1 };

vector<int> v1(t, t + 10);

deque<int> d1(t, t + 10);

set<int> s1(t, t + 10);

for_each(v1.begin(), v1.end(), myfunction); // Line I

for_each(d1.begin(), d1.end(), myfunction); // Line II

for_each(s1.begin(), s1.end(), myfunction); // Line III
```



```
return 0;
```

```
}
```

- A. program outputs: 10 5 9 6 2 4 7 8 3 1 10 5 9 6 2 4 7 8 3 1 1 2 3 4 5 6 7 8 9 10
- B. program outputs: 10 5 9 6 2 4 7 8 3 1 10 5 9 6 2 4 7 8 3 1 10 5 9 6 2 4 7 8 3 1
- C. program outputs: 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 10
- D. compilation error in line I
- E. compilation error in line III

**Answer: A**

#### QUESTION NO: 14

What happens when you attempt to compile and run the following code?

```
#include <iostream>
```

```
#include <algorithm>
```

```
#include <map>
```

```
using namespace std;
```

```
int main() {
```

```
int t[] = { 10, 5, 9, 6, 2, 4, 7, 8, 3, 1 };
```

```
map<int, int> m;
```

```
for(int i=0; i < 10; i++) {
```

```
    m[i]=t[i];
```

```
}
```

```
pair<const int,int> p(5,5);
```

```
map<int, int>::iterator it = find(m.begin(), m.end(), p);
```

```
if (it != m.end())
```

```
{
```

```
cout<<it?>first<<endl;

}

else

{

cout<<"Not found!\n";

}

return 0;

}
```

Program outputs:

- A. 5
- B. Not found!
- C. 10
- D. compilation error

**Answer: B**

### QUESTION NO: 15

What happens when you attempt to compile and run the following code?

```
#include <iostream>

#include <algorithm>

#include <vector>

#include <set>

using namespace std;

void myfunction(int i) {

cout << " " << i;

}
```

```
int main() {  
  
int t[] = { 10, 5, 9, 6, 2, 4, 7, 8, 3, 1 };  
  
set<int> s1(t, t+10);  
  
vector<int> v1(s1.rbegin(), s1.rend());  
  
swap_ranges(s1.begin(), s1.end(), v1.begin());  
  
for_each(v1.begin(), v1.end(), myfunction);  
  
for_each(s1.begin(), s1.end(), myfunction);  
  
return 0;  
  
}
```

Program outputs:

- A. 10 9 8 7 6 5 4 3 2 1 1 2 3 4 5 6 7 8 9 10
- B. compilation error
- C. 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 10
- D. 1 2 3 4 5 6 7 8 9 10 10 9 8 7 6 5 4 3 2 1
- E. 10 9 8 7 6 5 4 3 2 1 10 9 8 7 6 5 4 3 2 1

**Answer: B**

#### QUESTION NO: 16

What happens when you attempt to compile and run the following code?

```
#include <iostream>  
  
#include <set>  
  
#include <list>  
  
using namespace std;  
  
int main(){  
  
int t[]={ 1, 1, 2, 2, 3, 3, 4, 4, 5, 5 };
```

```
list<int>v(t, t+10);

set<int> s1(v.begin(),v.end());

if (s1.count(3) == 2) {

s1.erase(3);

}

for(set<int>::iterator i=s1.begin();i!= s1.end(); i++) {

cout<<*i<<" ";

}

return 0;

}
```

- A. program outputs: 1 2 3 4 5
- B. program outputs: 1 2 4 5
- C. program outputs: 1 1 2 2 3 4 4 5 5
- D. program outputs: 1 1 2 3 3 4 4 5 5
- E. compilation error

**Answer: A**

### QUESTION NO: 17

What happens when you attempt to compile and run the following code?

```
#include <vector>

#include <iostream>

#include <algorithm>

#include <functional>

using namespace std;

template<class T>struct Out {

ostream & out;
```

```
Out(ostream & o): out(o){}
```

```
void operator() (const T & val ) { out<<val<<" "; } };
```

```
int Add(int a, int b) {  
    return a+b;  
}
```

```
int main() {  
    int t[]={1,2,3,4,5,6,7,8,9,10};  
    vector<int> v1(t, t+10);  
    vector<int> v2(10);  
    transform(v1.begin(), v1.end(), v2.begin(), bind2nd(ptr_fun (Add),1));  
    vector<int>::iterator it = find_if(v2.begin(), v2.end(),bind2nd(equal_to<int>(),10));  
    cout<<*it<<endl;  
    return 0;  
}
```

Program outputs:

- A. false
- B. true
- C. 10
- D. 0
- E. compilation error

**Answer: C**

### QUESTION NO: 18

What happens when you attempt to compile and run the following code?

```
#include <iostream>

#include <algorithm>

#include <vector>

#include <deque>

using namespace std;

void myfunction(int i) {

cout << " " << i;

}

int main() {

int t[] = { 10, 5, 9, 6, 2, 4, 7, 8, 3, 1 };

deque<int> d1(t, t+10);

vector<int> v1(d1.rbegin(), d1.rend());

sort(d1.begin(), d1.end());

swap_ranges(v1.begin(), v1.end(), d1.begin());

for_each(v1.begin(), v1.end(), myfunction);

for_each(d1.begin(), d1.end(), myfunction);

return 0;

}
```

Program outputs:

- A. 10 9 8 7 6 5 4 3 2 1 1 2 3 4 5 6 7 8 9 10
- B. compilation error
- C. 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 10
- D. 1 2 3 4 5 6 7 8 9 10 1 3 8 7 4 2 6 9 5 10
- E. 1 3 8 7 4 2 6 9 5 10 1 2 3 4 5 6 7 8 9 10

**Answer: D**

**QUESTION NO: 19**

What happens when you attempt to compile and run the following code?

```
#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) { out<<val<<" "; } };

int main() {

int t1[]={3,2,4,1,5};

int t2[]={5,6,8,2,1};

vector<int> v1(10);

sort(t1, t1+5);

sort(t2, t2+5);

set_union(t1,t1+5,t2,t2+5,v1.begin());

for_each(v1.begin(), v1.end(), Out<int>(cout));cout<<endl;

return 0;

}
```

Program outputs:

**A.** 3 2 4 1 5 6 8 2 1 0

**B.** 1 2 3 4 5 6 8 2 1 0

- C. 1 1 2 2 3 4 5 5 6 8
- D. 1 2 3 4 5 6 8 0 0 0
- E. compilation error

**Answer: D**

### QUESTION NO: 20

What happens when you attempt to compile and run the following code?

```
#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) { out<<val<<" "; } };

int main() {

int t[]={3,2,4,1,5,10,9,7,8,6};

vector<int> v1(t,t+10);

sort(v1.begin(), v1.end(), greater<int>());

cout<<min_element(v1.begin(), v1.end());

return 0;

}
```

Program outputs:

- A. 3
- B. 1



- C. 6
- D. 10
- E. compilation error

**Answer: E**

### QUESTION NO: 21

What will happen when you attempt to compile and run the code below, assuming that you enter the following sequence: 1 2 3 end<enter>?

```
#include <iostream>

#include <string>

#include <list>

#include <algorithm>

using namespace std;

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) {out<<val<<" "; } };

int main ()

{

list<int> l;

for( ; !cin.bad() ; )

{

int i;
```

```
cin>>i;

l.push_back(i);

}

for_each(l.begin(), l.end(), Out<int>(cout));

return 0;

}
```

Program will output:

- A. 1 2 3
- B. 1 2 3 end
- C. 1
- D. compilation error
- E. program runs forever without output

**Answer: E**

## QUESTION NO: 22

What happens when you attempt to compile and run the following code?

```
#include <iostream>

#include <set>

#include <vector>

using namespace std;

template<class T> void print(T start, T end) {

while (start != end) {

std::cout << *start << " "; start++;

}

}

int main(){
```

```
vector<int>v;

multiset<int> s;

for(int i=10; i>0; i??) {

v.push_back(i); s.push_back(i);

}

print(v.begin(), v.end()); print(s.begin(), s.end());cout<<endl;

return 0;

}
```

- A. program outputs: 10 9 8 7 6 5 4 3 2 1 1 2 3 4 5 6 7 8 9 10
- B. program outputs: 10 9 8 7 6 5 4 3 2 1 10 9 8 7 6 5 4 3 2 1
- C. program outputs: 10 9 8 7 6 5 4 3 2 1 and unpredictable sequence of numbers range 1 to 10
- D. compilation error

**Answer: D**

### QUESTION NO: 23

What happens when you attempt to compile and run the following code?

```
#include <iostream>

#include <map>

using namespace std;

int main() {

int t[] = { 1, 1, 2, 2, 3, 3, 4, 4, 5, 5 };

string s[] = { "one", "one", "two", "two", "three", "three", "four", "four", "five", "five"};

map<int, string> m;

for (int i = 0; i < 10; i++) {

m.push_back(pair<int, string>(t[i], s[i]));
```

```
}  
  
for (map<int, string>::iterator i = m.begin(); i != m.end(); i++) {  
    cout << i->first << " ";  
}  
  
return 0;  
}
```

- A. program outputs: 1 2 3 4 5
- B. compilation error
- C. program outputs: 1 1 2 2 3 3 4 4 5 5
- D. program outputs: one two three four five
- E. program outputs: one one two two three three four four five five

**Answer: B**

#### QUESTION NO: 24

What happens when you attempt to compile and run the following code?

```
#include <deque>  
  
#include <iostream>  
  
#include <algorithm>  
  
using namespace std;  
  
class B { int val;  
  
public:  
  
    B(int v):val(v){}  
  
    int getV() const {return val;}  bool operator < (const B & v) const { return val<v.val;} };  
  
ostream & operator <<(ostream & out, const B & v) { out<<v.getV(); return out;}  
  
template<class T>struct Out {
```

```
ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) { out<<val<<" "; } };

int main() {

int t[]={8, 10, 5, 1, 4, 6, 2, 7, 9, 3};

deque<B> d1(t, t+10);

sort(d1.begin(), d1.end());

deque<B>::iterator it = upper_bound(d1.begin(), d1.end(), B(4), greater<B>());

for_each(it, d1.end(), Out<B>(cout)); cout<<endl;

return 0;

}
```

Program outputs:

- A. 5 6 7 8 9 10
- B. 4 5 6 7 8 9 10
- C. compilation error
- D. 1 2 3 4 5
- E. 1 2 3 4

**Answer: C**

### QUESTION NO: 25

Which stack initialization (line numbers) are correct? Choose all that apply.

```
#include <iostream>

#include <deque>

#include <list>

#include <stack>

#include <vector>
```

```
using namespace std;
```

```
int main()
{
deque<int> mydeck;

list<int> mylist;

vector<int> myvector;

stack<int> first;// Line I

stack<int> second(mydeck);// Line II

stack<int> third(second);// Line III

stack<int, list<int> > fourth(mylist);// Line IV

stack<int, vector<int> > fifth(myvector);// Line V

return 0;

}
```

- A. line I
- B. line II
- C. line III
- D. line IV
- E. line V

**Answer: A,B,C,D,E**

#### **QUESTION NO: 26**

What happens when you attempt to compile and run the following code?

```
#include <deque>

#include <iostream>

#include <algorithm>
```

```
using namespace std;

class B { int val;

public:

B(int v):val(v){}

int getV() const {return val;} bool operator < (const B & v) const { return val<v.val;} };

ostream & operator <<(ostream & out, const B & v) { out<<v.getV(); return out;}

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) { out<<val<<" "; } };

int main() {

int t[]={8, 10, 5, 1, 4, 6, 2, 7, 9, 3};

deque<B> d1(t, t+10);

sort(d1.begin(), d1.end());

deque<B>::iterator it = upper_bound(d1.begin(), d1.end(), B(4));

for_each(it, d1.end(), Out<B>(cout)); cout<<endl;

return 0;

}
```

Program outputs:

- A. 5 6 7 8 9 10
- B. 4 5 6 7 8 9 10
- C. 6 7 8 9 10
- D. 1 2 3 4 5
- E. 1 2 3 4

**Answer: A**

**QUESTION NO: 27**

What happens when you attempt to compile and run the following code?

```
#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

class B { int val;

public:

B(int v):val(v){}

int getV() const {return val;} bool operator < (const B & v) const { return val<v.val;} };

ostream & operator <<(ostream & out, const B & v) { out<<v.getV(); return out;}

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) { out<<val<<" "; } };

int main() {

B t1[]={3,2,4,1,5};

B t2[]={5,6,8,2,1};

vector<B> v1(10,0);

sort(t1, t1+5);

sort(t2, t2+5);

set_symmetric_difference(t2,t2+5,t1,t1+5,v1.begin());

for_each(v1.begin(), v1.end(), Out<B>(cout));cout<<endl;

return 0;

}
```



Program outputs:

- A. 6 8 3 4 0 0 0 0 0 0
- B. 3 4 0 0 0 0 0 0 0 0
- C. 6 8 0 0 0 0 0 0 0 0
- D. compilation error
- E. 3 4 6 8 0 0 0 0 0 0

**Answer: E**

### QUESTION NO: 28

What happens when you attempt to compile and run the following code?

```
#include <vector>

#include <iostream>

#include <algorithm>

#include <functional>

using namespace std;

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) { out<<val<<" "; } };

int main() {

int t[]={3,2,4,1,5,6,10,8,7,9};

vector<int> v1(t, t+10);

for_each(v1.begin(), v1.end(), bind2nd(plus<int>(), 1));

for_each(v1.rbegin(), v1.rend(), Out<int>(cout));cout<<endl;

return 0;
```

```
}
```

Program outputs:

- A. 3 2 4 1 5 6 10 8 7 9
- B. 4 3 5 2 6 7 11 9 8 10
- C. 9 7 8 10 6 5 1 4 2 3
- D. 10 8 9 11 7 6 2 5 3 4
- E. compilation error

**Answer: C**

### QUESTION NO: 29

What happens when you attempt to compile and run the following code?

```
#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

class B { int val;

public:

B(int v=0):val(v){}

int getV() const {return val;}

operator int () const { return val; } };

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) { out<<val<<" "; } };

struct Add {
```

```
B operator()(B & a, B & b) { return a+b; };
```

```
int main() {
```

```
int t[]={1,2,3,4,5,6,7,8,9,10};
```

```
vector<B> v1(t, t+10);
```

```
vector<B> v2(10);
```

```
transform(v1.begin(), v1.end(), v2.begin(), bind1st(1,Add()));
```

```
for_each(v2.rbegin(), v2.rend(), Out<B>(cout));cout<<endl;
```

```
return 0;
```

```
}
```

Program outputs:

- A. 1 2 3 4 5 6 7 8 9 10
- B. 2 3 4 5 6 7 8 9 10 11
- C. 10 9 8 7 6 5 4 3 2 1
- D. 11 10 9 8 7 6 5 4 3 2
- E. compilation error

**Answer: E**

### QUESTION NO: 30

What happens when you attempt to compile and run the following code?

```
#include <iostream>
```

```
#include <algorithm>
```

```
#include <vector>
```

```
using namespace std;
```

```
int main () {
```

```
int t[] = {1,2,3,2,3,5,1,2,7,3,2,1,10, 4,4,5};
```

```
vector<int> v (t,t+15);

int number = count(v.begin(), v.end(), 2);

cout<< number<<endl;

return 0;

}
```

Program outputs:

- A. 4
- B. 3
- C. 2
- D. 0
- E. compilation error

**Answer: A**

### QUESTION NO: 31

What happens when you attempt to compile and run the following code?

```
#include <deque>

#include <iostream>

#include <algorithm>

#include <set>

using namespace std;

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) { out<<val<<" "; }

};
```

```
bool Compare(char a, char b) { return tolower(a) < tolower(b);}

int main() {

char s[]{"qwerty"};

char t1[]{"ert"};

char t2[]{"ERT"};

sort(s, s+6);

cout<<includes(s,s+6, t1,t1+3, Compare)<<" " <<includes(s,s+6, t2,t2+3, Compare)<<endl;

return 0;

}
```

Program outputs:

- A. 0 0
- B. 0 1
- C. 1 0
- D. 1 1

**Answer: D**

### QUESTION NO: 32

What happens when you attempt to compile and run the following code?

```
#include <vector>

using namespace std;

int main ()

{

std::vector<int>v1;

v1.push_back(10);

return 0;

}
```

- A. compilation fails due to error in line 2
- B. compilation fails due to error in line 5
- C. exception is thrown during run time
- D. code compiles and executes successfully

**Answer: D**

### QUESTION NO: 33

What will happen when you attempt to compile and run the following code?

```
#include <iostream>

#include <set>

#include <vector>

using namespace std;

int main(){

int t[]={ 3, 4, 2, 1, 6, 5, 7, 9, 8, 0 };

vector<int>v(t, t+10);

set<int> s1(v.begin(),v.end());

s1.insert(v.begin(),v.end());

bool found = s1.find(7);

if (found){

cout<<"Element found!\n";

}else {

cout<<"Element not found!\n";

}

return 0;

}
```

- A. program will display "Element found!"

- B. program will display "Element not found!\n"
- C. code will not compile
- D. changing type of variable found to int will make this code compile

**Answer: C**

### QUESTION NO: 34

What happens when you attempt to compile and run the following code?

```
#include <iostream>

#include <algorithm>

#include <vector>

#include <set>

using namespace std;

void myfunction(int i) {

cout << " " << i;

}

bool classifier(int v) {

return v%2==0;

}

int main() {

int t[] = { 1, 5, 2, 5, 2, 4, 4, 3, 3, 1 };

vector<int> v1(t, t+10);

set<int> s1(t, t+10);

replace(v1.begin(), v1.end(), classifier, 10);

for_each(v1.begin(), v1.end(), myfunction);

return 0;
```

```
}
```

Program outputs:

- A. 1 5 10 5 10 10 10 3 3 1
- B. 1 5 2 5 2 4 4 3 3 1
- C. compilation error
- D. 10 10 2 10 2 4 4 10 10 10

**Answer: C**

### QUESTION NO: 35

What happens when you attempt to compile and run the following code?

```
#include <deque>

#include <list>

#include <iostream>

using namespace std;

int main ()

{

list<int>l1;

deque<int>d1;

for(int i=0; i<5; i++)

{

l1.push_back(i);l1.push_front(i);

d1.push_back(i);d1.push_front(i);

}

for(int i=0; i<d1.size(); i++)

{

cout<<d1[i]<<" "<<l1[i]<<" ";
```



```
}  
  
cout<<endl;  
  
return 0;  
  
}
```

- A. program displays 4 4 3 3 2 2 1 1 0 0 0 0 1 1 2 2 3 3 4 4
- B. runtime exception
- C. compilation error due to line 11
- D. compilation error due to line 12
- E. compilation error due to line 16

**Answer: E**

### QUESTION NO: 36

What happens when you attempt to compile and run the following code?

```
#include <deque>  
  
#include <vector>  
  
#include <iostream>  
  
using namespace std;  
  
int main ()  
{  
int t[] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };  
vector<int> v1(t, t + 10);  
deque<int> d1(v1.begin(), v1.end());  
deque<int> d2;  
d2 = d1;  
d2.insert(d1.rbegin(), 10);
```

```
for(int i = 0; i<d1.size(); i++)  
  
{  
  
cout<<d1[i]<<" ";  
  
}  
  
return 0;  
  
}
```

- A. program outputs: 0 1 2 3 4 5 6 7 8 9 10
- B. program outputs: 10 0 1 2 3 4 5 6 7 8 9
- C. program outputs: 0 1 2 3 4 5 6 7 8 9
- D. compilation error

**Answer: D**

### QUESTION NO: 37

What happens when you attempt to compile and run the following code?

```
#include <iostream>  
  
using namespace std;  
  
int main()  
  
{  
  
cout<<100<<" ";  
  
cout.setf(ios::hex);  
  
cout<<100<<" ";  
  
return 0;  
  
}
```

Program outputs:

- A. 100 64

- B. 100 0x64
- C. 0x64 0x64
- D. 64 0x64
- E. 100 100

**Answer: E**

### QUESTION NO: 38

What happens when you attempt to compile and run the following code?

```
#include <iostream>

using namespace std;

int main()
{
    cout.setf(ios::hex, ios::basefield);

    cout<<100<<" ";

    cout.flags(ios::showbase);

    cout<<100<<" ";

    return 0;
}
```

Program outputs:

- A. 64 64
- B. 64 0x64
- C. 0x64 0x64
- D. 64 100
- E. compilation error

**Answer: D**

**QUESTION NO: 39**

What happens when you attempt to compile and run the following code?

```
#include <iostream>

#include <algorithm>

#include <vector>

using namespace std;

int main () {

int t[] = {1,2,3,2,3,5,1,2,7,3,2,1,10, 4,4,5};

vector<int> v (t,t+15);

vector<int>::iterator it = search_n(v.begin(), v.end(), 4, 2);

cout<< it?v.begin()<<endl;

return 0;

}
```

Program outputs:

- A. 10
- B. 3
- C. 1
- D. 15
- E. compilation error

**Answer: D**

**QUESTION NO: 40**

What happens when you attempt to compile and run the following code?

```
#include <deque>

#include <vector>

#include <iostream>

#include <string>

using namespace std;

template<typename T>

void print(T start, T end)

{

while (start != end)

cout<<*start++;

}

int main ()

{

string t[] = {"one", "two", "three", "four", "five"};

vector<string>v1(t, t+5);

deque<string>d1(v1.rbegin(), v1.rend());

d1.push_back("zero");

print(d1[0].rbegin(),d1[0].rend());

return 0;

}
```

- A. program outputs: orez
- B. program outputs: evif
- C. compilation error
- D. program outputs: five

**Answer: B**

**QUESTION NO: 41**

Which pieces of code inserted independently into places marked 1 and 2 will cause the program to compile and display: 0 1 2 3 4 5 6 7 8 9? Choose all that apply.

```
#include <list>

#include <iostream>

using namespace std;

class A { int a; public:
A(int a){ this->a=a;}

//insert code here 1

};

//insert code here 2

template<class T> void print(T start, T end) {
while (start != end) {
std::cout << *start << " "; start++;
}
}

int main() {
A t1[] = { 1, 7, 8, 4, 5 };list<A> l1(t1, t1 + 5);
A t2[] = { 3, 2, 6, 9, 0 };list<A> l2(t2, t2 + 5);
l1.sort();l2.sort();l1.merge(l2);
print(l1.begin(), l1.end());
print(l2.begin(), l2.end()); cout<<endl;
return 0;
}
```

**A.** place 1: operator int() { return a; }

**B.** place 1: operator int() { return a; }

bool operator < (const A & b) { return this->a< b.a;}

**C.** place 1: bool operator < (const A & b) { return this->a< b.a;}

**D.** place 1: `bool operator < (const A & b) { return this?>a< b.a;}`  
friend ostream & operator <<(ostream & c, const A & a);  
place 2: `ostream & operator <<(ostream & c, const A & a) { c<<a.a; return c;}`  
**E.** place 1: `bool operator < (const A & b) { return this?>a< b.a;}`  
place 2: `ostream & operator <<(ostream & c, const A & a) { c<<a.a; return c;}`

**Answer: A,B,D**

## QUESTION NO: 42

What happens when you attempt to compile and run the following code?

```
#include <iostream>

#include <algorithm>

#include <vector>

using namespace std;

int main () {

int t[] = {1,2,3,4,5,1,2,3,4,5};

vector<int> v (t,t+10);

vector<int>::iterator it;

int m1[] = {1, 2, 3};

it = search (v.begin(), v.end(), m1, m1+3);

cout << "found at position: " << it?v.begin() << endl;

return 0;

}
```

Program outputs:

- A.** found at position: 5
- B.** found at position: 0
- C.** found at position: 6

- D. found at position: 1
- E. found at position: 10

**Answer: B**

### QUESTION NO: 43

What happens when you attempt to compile and run the following code?

```
#include <deque>

#include <iostream>

#include <algorithm>

using namespace std;

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) { out<<val<<" "; } };

struct Sequence { int start;

Sequence(int start):start(start){}

int operator()() {return 10*(1+(start++ %3));}

};

int main() {

deque<int> d1(10);

generate(d1.begin(), d1.end(), Sequence(1));

sort(d1.begin(), d1.end());

pair<deque<int>::iterator, deque<int>::iterator > result = equal_range(d1.begin(), d1.end(), 20);

for_each(result.first, result.second, Out<int>(cout));cout<<endl;

return 0;

}
```



Program outputs:

- A. 10 10 10 20 20 20 20 30 30 30
- B. 20 20 20 20
- C. 10 20 20 20 20
- D. 20 20 20 20 30
- E. 10 20 20 20 20 30

**Answer: B**

#### QUESTION NO: 44

Which changes, introduced independently, will allow the code to compile and display “one” “eight” “nine” “ten”? Choose all that apply

```
#include <iostream>

#include <map>

#include <string>

using namespace std;

class A {

int a;

public:

A(int a):a(a){}

int getA() const { return a;}

/* Insert Code Here 1 */

};

/* Insert Code Here 2 */

int main(){

int t[]={ 3, 4, 2, 1, 6, 5, 7, 9, 8, 10 };

string s[] = {"three", "four", "two", "one", "six", "five", "seven", "nine", "eight", "ten"};
```

```

map<A, string> m; /* Replace Code Here 3 */

for(int i=0; i<10; i++) {

m.insert(pair<A,string>(A(t[i]),s[i]));

}

m.erase(m.lower_bound(2),m.upper_bound(7));

map<A, string>::iterator i=m.begin(); /* Replace Code Here 4 */

for( ;i!= m.end(); i++) {

cout<<i?>second<<" ";

}

cout<<endl;

return 0;

}

```

- A. operator int() const { return a;} inserted at Place 1
- B. bool operator < (const A & b) const { return a<b.a;} inserted at Place 1
- C. bool operator < (const A & b) const { return b.a<a;} inserted at Place 1
- D. struct R { bool operator ()(const A & a, const A & b) { return a.getA()<b.getA();} }; inserted at Place 2

replacing line marked 3 with map<A, string, R> m;

replacing line marked 4 with map<A, string,R>::iterator i=m.begin();

**Answer: A,B,D**

#### QUESTION NO: 45

What happens when you attempt to compile and run the following code?

```

#include <vector>

#include <iostream>

#include <algorithm>

#include <functional>

```

```
using namespace std;

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) { out<<val<<" "; } };

int main() {

int t1[]={3,2,4,1,5};

int t2[]={6,10,8,7,9};

vector<int> v1(5);

transform(t1,t1+5,t2,v1.rbegin(), plus<int>());

for_each(v1.rbegin(), v1.rend(), Out<int>(cout));cout<<endl;

return 0;

}
```

Program outputs:

- A. 9 12 12 8 14
- B. 14 8 12 12 9
- C. 3 2 4 1 5 6 10 8 7 9
- D. 1 2 3 4 5 6 7 8 9 10
- E. compilation error

**Answer: A**

#### QUESTION NO: 46

What happens when you attempt to compile and run the following code?

```
#include <vector>

#include <iostream>
```

```
#include <algorithm>

using namespace std;

class B { int val;

public:

B(int v):val(v){}

int getV() const {return val;} bool operator < (const B & v) const { return val<v.val; } };

ostream & operator <<(ostream & out, const B & v) { out<<v.getV(); return out;}

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) { out<<val<<" "; } };

int main() {

B t1[]={3,2,4,1,5};

int t2[]={5,6,8,2,1};

vector<B> v1(10,0);

sort(t1, t1+5);

sort(t2, t2+5);

set_union(t1,t1+5,t2,t2+5,v1.begin());

for_each(v1.begin(), v1.end(), Out<B>(cout));cout<<endl;

return 0;

}
```

Program outputs:

- A. 3 2 4 1 5 6 8 2 1 0
- B. 1 2 3 4 5 6 8 2 1 0
- C. 1 1 2 2 3 4 5 5 6 8
- D. 1 2 3 4 5 6 8 0 0 0
- E. compilation error

**Answer: E**

**QUESTION NO: 47**

What happens when you attempt to compile and run the following code?

```
#include <iostream>

#include <map>

#include <vector>

#include <sstream>

#include <string>

using namespace std;

int main(){

int t[] ={ 3, 4, 2, 1, 6, 5, 7, 9, 8, 0 };

vector<int> v(t, t+10);

map<int,string> m;

for(vector<int>::iterator i=v.begin(); i!=v.end(); i++) {

stringstream s; s<<*i<<*i; m.insert(pair<int,string>(*i,s.str()));

}

for(map<int, string>::iterator i=m.begin();i!= m.end(); i++) {

cout<<*i<<" ";

}

return 0;

}
```

- A.** program outputs: 3 4 2 1 6 5 7 9 8 0
- B.** program outputs: 00 11 22 33 44 55 66 77 88 99
- C.** program outputs: 0 1 2 3 4 5 6 7 8 9
- D.** program outputs: 0 00 1 11 2 22 3 33 4 44 5 55 6 66 7 77 8 88 9 99

E. compilation error

**Answer: E**

**QUESTION NO: 48**

What will happen when you attempt to compile and run the code below, assuming that file test.in contains the following sequence: 1 2 3?

```
#include <iostream>

#include <fstream>

#include <string>

#include <list>

#include <algorithm>

using namespace std;

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) {out<<val<<" "; } };

int main () {

ifstream f("test.in");

list<int> l;

for( ; !f.fail() ; ) {

int i;

f>>i;

l.push_back(i);

}

f.close();
```

```
for_each(l.begin(), l.end(), Out<int>(cout));  
  
return 0;  
  
}
```

Program will output:

- A. 1 2 3
- B. 1 2 3 3
- C. no output
- D. compilation error
- E. program runs forever without output

**Answer: B**

#### QUESTION NO: 49

What happens when you attempt to compile and run the following code?

```
#include <vector>  
  
#include <iostream>  
  
#include <algorithm>  
  
using namespace std;  
  
template<class T>struct Out {  
    ostream & out;  
  
    Out(ostream & o): out(o){}  
  
    void operator() (const T & val ) { out<<val<<" "; }  
};  
  
int main() {  
  
    int t1[]={3,2,4,1,5};  
  
    int t2[]={6,10,8,7,9};  
  
    vector<int> v1(10);
```

```
sort(t1, t1+5); sort(t2, t2+5);  
copy(t1,t1+5,v1.begin());  
copy(t2,t2+5,v1.begin()+5);  
merge(v1.begin(), v1.begin()+5,v1.end());  
for_each(v1.begin(), v1.end(), Out<int>(cout));cout<<endl;  
return 0;  
}
```

Program outputs:

- A. 1 2 3 4 5 6 10 8 7 9
- B. 3 2 4 1 5 6 7 8 9 10
- C. 3 2 4 1 5 6 10 8 7 9
- D. 1 2 3 4 5 6 7 8 9 10
- E. compilation error

**Answer: E**

### QUESTION NO: 50

What happens when you attempt to compile and run the following code?

```
#include <vector>  
#include <set>  
#include <iostream>  
#include <algorithm>  
  
using namespace std;  
  
void print(int v) { cout<<v<<" "; }  
struct Sequence {
```



```
int start;

Sequence(int start):start(start){}

int operator()() { return start++; }

};

bool predicate(int v) { return v%2==0; }

int main() {

vector<int> v1(10);

generate_n(v1.begin(), 10, Sequence(1));

set<int> s1(v1.begin(), v1.end());

remove_if(s1.begin(), s1.end(), predicate);

for_each(s1.begin(), s1.end(), print);cout<<endl;

return 0;

}
```

Program outputs:

- A. 1 3 5 7 9 6 7 8 9 10
- B. 1 3 5 7 9
- C. 2 4 6 8 10
- D. compilation error

**Answer: D**

#### QUESTION NO: 51

What happens when you attempt to compile and run the following code?

```
#include <string>

#include <list>

#include <iostream>

using namespace std;
```

```
template<class T> void print(T start, T end) {  
  
while (start != end) {  
  
std::cout << *start << " "; start++;  
  
}  
  
}  
  
  
int main() {  
  
string t1[] ={"1", "2", "3", "4", "5", "6", "7", "8", "9", "10"};  
  
list<string> l1(t1, t1 + 10);  
  
list<string> l2(l1);  
  
l2.reverse(); l1.splice(l1.end(),l2);  
  
l1.unique();  
  
print(l1.begin(), l1.end()); cout<<endl;  
  
return 0;  
  
}
```

- A. compilation error
- B. program outputs: 1 2 3 4 5 6 7 8 9 10 9 8 7 6 5 4 3 2 1
- C. program outputs: 1 2 3 4 5 6 7 8 9 10 10 9 8 7 6 5 4 3 2 1
- D. program outputs: 1 2 3 4 5 6 7 8 9 10

**Answer: B**

### QUESTION NO: 52

What happens when you attempt to compile and run the following code?

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

```
using namespace std;

int main () {
int t[] = {1,2,3,4,5,1,2,3,5,4};
vector<int> v (t,t+10);
vector<int>::iterator it;
int m1[] = {1, 3, 2};
it = find_end (v.begin(), v.end(), m1, m1+3);
if (it != v.end())
cout << "Found at position: " << it?v.begin() << endl;
return 0;
}
```

- A. program outputs: Found at position: 5
- B. program outputs: Found at position: 0
- C. no output
- D. program outputs: Found at position: 10

**Answer: C**

### QUESTION NO: 53

What will happen when you attempt to compile and run the code below, assuming that file test.in contains the following sequence: 1 2 3?

```
#include <iostream>
#include <fstream>
#include <string>
#include <list>
#include <algorithm>
```

```
using namespace std;

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) {out<<val<<" "; } };

int main () {

ifstream f("test.in");

list<int> l;

for( ; f.good() ; ) {

int i;

f>>i;

l.push_back(i);

}

f.close();

for_each(l.begin(), l.end(), Out<int>(cout));

return 0;

}
```

Program will output:

- A. 1 2 3
- B. 1 2 3 3
- C. no output
- D. compilation error
- E. program runs forever without output

**Answer: A**

**QUESTION NO: 54**

What happens when you attempt to compile and run the following code?

```
#include <vector>

#include <iostream>

int main ()
{
int t[]={1,2,3,4,5};

std::vector<int>v1(t,t+5);

std::vector<int>v2(v1);

v1.resize(10);

v2.reserve(10);

std::vector<int>::iterator i = v1.begin();int ii = 0;

while (i != v1.end()) { std::cout<<i[ii]<<" ";ii++;i++; }

i = v2.begin();ii=0;

while (i != v2.end()) { std::cout<<i[ii]<<" ";ii++;i++; }

return 0;

}
```

- A. program outputs 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
- B. compilation error
- C. program outputs 1 1 1 1 1 1 1 1 1 1 1 2 3 4 5
- D. program outputs 1 2 3 4 5 0 0 0 0 0 1 2 3 4 5 0 0 0 0 0

**Answer: A**

**QUESTION NO: 55**

What will happen when you attempt to compile and run the following code?

```
#include <iostream>
```

```
using namespace std;
```

```
template <class T>
```

```
class A {
```

```
    T _v;
```

```
public:
```

```
    A(T v);
```

```
};
```

```
template<class T>
```

```
A. :A(T v):_v(v) {}
```

```
int main()
```

```
{
```

```
    A<int> a(2);
```

```
    cout<<1<<endl;
```

```
    return 0;
```

```
}
```

**B.** program will display: 1

**C.** program will not compile

**D.** program will compile

**E.** program will cause runtime exception

**Answer: B**

## QUESTION NO: 56

What happens when you attempt to compile and run the following code?

```
#include <vector>
```

```
#include <iostream>
```

```
#include <algorithm>

#include <functional>

using namespace std;

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) { out<<val<<" "; } };

int main() {

int t1[]={1,2,3,4,5,6,7,8,9,10};

int t2[]={1,2,3,4,5,6,7,8,9,10};

vector<int> v1(t1, t1+10);

vector<int> v2(t2, t2+10);

vector<int> v3(10);

transform(v1.begin(), v1.end(), v2.rbegin(), v3.begin(), minus<int>());

for_each(v3.rbegin(), v3.rend(), Out<int>(cout));cout<<endl;

return 0;

}
```

Program outputs:

- A. 9 7 5 3 1 ?1 ?3 ?5 ?7 ?9
- B. ?1 ?3 ?5 ?7 ?9 9 7 5 3 1
- C. 1 3 5 7 9 ?1 ?3 ?5 ?7 ?9
- D. 1 3 5 7 9 ?1 ?3 ?5 ?7 ?9
- E. ?9 ?7 ?5 ?3 ?1 1 3 5 7 9

**Answer: A**

**QUESTION NO: 57**

What happens when you attempt to compile and run the following code?

```
#include <vector>

#include <iostream>

#include <algorithm>

#include <functional>

using namespace std;

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) { out<<val<<" "; };

}

struct Add : public binary_function<int, int, int> {

int operator() (const int & a, const int & b) const {

return a+b;

}

};

int main() {

int t[]={1,2,3,4,5,6,7,8,9,10};

vector<int> v1(t, t+10);

vector<int> v2(10);

transform(v1.begin(), v1.end(), v2.begin(), bind1st(Add(), 1));

for_each(v2.rbegin(), v2.rend(), Out<int>(cout));cout<<endl;

return 0;

}
```

Program outputs:



- A. 1 2 3 4 5 6 7 8 9 10
- B. 2 3 4 5 6 7 8 9 10 11
- C. 10 9 8 7 6 5 4 3 2 1
- D. 11 10 9 8 7 6 5 4 3 2
- E. compilation error

**Answer: D**

### QUESTION NO: 58

Which changes introduced independently will allow the code to compile and display “one” “eight” “nine” “ten”? Choose all that apply.

```
#include <iostream>

#include <map>

#include <string>

using namespace std;

class A {

int a;

public:

A(int a):a(a){}

int getA() const { return a;}

/* Insert Code Here 1 */

};

/* Insert Code Here 2 */

int main(){

int t[]={ 3, 4, 2, 1, 6, 5, 7, 9, 8, 10 };

string s[] = {"three", "four", "two", "one", "six", "five", "seven", "nine", "eight", "ten"};

multimap<A, string> m; /* Replace Code Here 3 */

for(int i=0; i<10; i++) {
```

```
m.insert(pair<A,string>(A(t[i]),s[i]));  
  
}  
  
m.erase(m.lower_bound(2),m.upper_bound(7));  
  
multimap<A, string>::iterator i=m.begin();/* Replace Code Here 4 */  
  
for( ; i!= m.end(); i++) {  
  
cout<<i?>second<<" "  
  
}  
  
cout<<endl;  
  
return 0;  
  
}
```

- A. operator int() const { return a;} inserted at Place 1
  - B. bool operator < (const A & b) const { return a<b.a;} inserted at Place 1
  - C. bool operator < (const A & b) const { return b.a<a;} inserted at Place 1
  - D. struct R { bool operator()(const A & a, const A & b) { return a.getA()<b.getA();} }; inserted at Place 2
- replacing line marked 3 with multimap<A, string, R> m;  
replacong line marked 4 with multimap<A, string, R>::iterator i=m.begin();

**Answer: A,B,D**

### QUESTION NO: 59

What happens when you attempt to compile and run the following code?

```
#include <iostream>  
  
#include <algorithm>  
  
#include <map>  
  
using namespace std;  
  
int main() {
```

```
int t[] = { 10, 5, 9, 6, 2, 4, 7, 8, 3, 1 };  
  
map<int, int> m;  
  
for(int i=0; i < 10; i++) {  
    m[i]=t[i];  
}  
  
map<int, int>::iterator it = find(m.begin(), m.end(), 5);  
  
cout<<it->first;  
  
return 0;  
}
```

Program outputs:

- A. 5
- B. 4
- C. 10
- D. compilation error

**Answer: D**

### QUESTION NO: 60

What happens when you attempt to compile and run the following code? Choose all that apply.

```
#include <iostream>  
  
#include <fstream>  
  
#include <string>  
  
#include <list>  
  
#include <algorithm>  
  
#include <iomanip>  
  
using namespace std;  
  
template<class T>struct Out {
```

```
ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) {out<<setw(3)<<hex<<val; } };

int main () {

int t[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

fstream f("test.out", ios::trunc|ios::out);

list<int> l(t, t+10);

for_each(l.begin(), l.end(), Out<int>(f));

f.close(); f.open("test.out");

for( ; f.good() ; ) {

int i; f>>i;

cout<<i<<" ";

}

f.close();

return 0;

}
```

- A. file test.out will be opened writing
- B. file test.out will be truncated
- C. file test.out will be opened for reading
- D. no file will be created nor opened
- E. program will display sequence 1 2 3 4 5 6 7 8 9 10

**Answer: A,B,C,E**

#### QUESTION NO: 61

What happens when you attempt to compile and run the following code?

```
#include <iostream>
```

```
#include <string>

using namespace std;

template <class T>
class A {
    T _v;
public:
    A() {}
    A(T v): _v(v){}
    T getV() { return _v; }

    void add(T & a) { _v+=a; }

    void add(string & a) {
        _v.insert(0, a);
    }

};

int main()
{
    A<string>a("Hello");
    string s(" world!");
    a.add(s);
    cout << a.getV() <<endl;
    return 0;
}
```

```
}
```

- A. program will display: Hello world!
- B. compilation error
- C. program will display: world!Hello
- D. program will run without any output

**Answer: B**

### QUESTION NO: 62

What happens when you attempt to compile and run the following code?

```
#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator()(const T & val ) {

out<<val<<" "; }

struct Sequence {

int start;

Sequence(int start):start(start){}

int operator()() {

return start++; } };

int main() {

vector<int> v1(10);
```

```
vector<int> v2(10);  
  
generate(v1.begin(), v1.end(), Sequence(1));  
  
reverse_copy(v1.begin(),v1.end(), v2.rbegin());  
  
sort(v2.begin(), v2.end(), less_equal<int>());  
  
for_each(v2.begin(), v2.end(), Out<int>(cout) );cout<<endl;  
  
return 0;  
  
}
```

Program outputs:

- A. 1 2 3 4 5 6 7 8 9 10
- B. 10 9 8 7 6 5 4 3 2 1
- C. no output
- D. compilation error

**Answer: A**

### QUESTION NO: 63

What happens when you attempt to compile and run the following code?

```
#include <iostream>  
  
#include <algorithm>  
  
#include <set>  
  
using namespace std;  
  
class A {  
  
int a;  
  
public:  
  
A(int a) : a(a) {}  
  
int getA() const { return a; } void setA(int a) { this->a = a; }  
  
bool operator < (const A & b) const { return a<b.a;}  
  
}
```

```
};  
  
struct Compare {  
  
bool operator()(A & a) {  
  
    if (a.getA() < 5) return true;  
  
    return false;  
  
}  
  
};  
  
int main () {  
  
int t[] = {1,2,3,2,3,5,1,2,7,3,2,1,10, 4,4,5};  
  
set<A> d (t,t+15);  
  
int number = count_if(d.begin(), d.end(), Compare());  
  
cout<< number<<endl;  
  
return 0;  
  
}
```

Program outputs:

- A. 12
- B. 4
- C. 2
- D. 0
- E. compilation error

**Answer: E**

#### **QUESTION NO: 64**

Given three files: class.h, class.cpp and main.cpp containing small C++ project, which sentences are TRUE if you attempt to compile and run the program? Assume that the whole compiling environment is properly set.

```
// File: main.cpp
```



```
#include <iostream>

#include "class.h"

using namespace std;

int main()
{
A<int> a;

cout << a.getV() << endl;

return 0;
}
```

```
//File: class.h

#ifndef _CLASS_

#define _CLASS_

template <class T>

class A {

T_v;

public:

A() {}

A(T v);

T getV();

};

#endif
```

```
//File: class.cpp

#include "class.h"
```

```
template<typename T>
```

```
A<T>::A(T v):_v(v) {}
```

```
template<class T>
```

```
T A<T>::getV() { return _v; }
```

- A. program will display: 0
- B. program will not compile
- C. program will display unpredictable number
- D. program will be not linked

**Answer: D**

#### QUESTION NO: 65

What happens when you attempt to compile and run the following code?

```
#include <iostream>
```

```
using namespace std;
```

```
template<class A>
```

```
void f(A a)
```

```
{
```

```
cout<<1<<endl;
```

```
}
```

```
void f(int a)
```

```
{
```

```
cout<<2<<endl;
```

```
}  
  
int main()  
{  
int a = 1;  
f<float>(a);  
return 0;  
}
```

- A. program displays: 1
- B. program displays: 2
- C. compilation error
- D. runtime exception

**Answer: A**

#### **QUESTION NO: 66**

What happens when you attempt to compile and run the following code?

```
#include <list>  
  
#include <iostream>  
  
using namespace std;  
  
template<class T>  
void print(T start, T end) {  
while (start != end) {  
std::cout << *start << " "; start++;  
}  
}  
  
int main()
```

```
{  
int t1[] = { 1, 7, 8, 4, 5 };  
list<int> l1(t1, t1 + 5);  
int t2[] = { 3, 2, 6, 9, 0 };  
list<int> l2(t2, t2 + 5);  
l1.sort();  
list<int>::iterator it = l2.begin();  
it++; it++;  
l1.splice(l1.end(), l2, it, l2.end());  
print(l1.begin(), l1.end()); cout<<"Size:"<<l1.size()<<" ";  
print(l2.begin(), l2.end()); cout<<"Size:"<<l2.size()<<endl;  
return 0;  
}
```

- A. program outputs: 1 4 5 7 8 6 9 0 Size:8 3 2 Size:2
- B. program outputs: 1 4 5 7 8 6 9 0 Size:8 3 2 6 9 0 Size:5
- C. compilation error
- D. program outputs: 0 1 4 5 6 7 8 9 Size:8 3 2 Size:2
- E. program outputs: 0 1 4 5 6 7 8 9 Size:8 3 2 6 9 0 Size:5

**Answer: A**

### QUESTION NO: 67

What happens when you attempt to compile and run the following code?

```
#include <vector>  
  
#include <iostream>  
  
#include <algorithm>  
  
using namespace std;
```

```
template<class T>struct Out {
ostream & out;
Out(ostream & o): out(o){}
void operator()(const T & val ) {
    out<<val<<" ";
}
};

struct Sequence {
int start;
Sequence(int start):start(start){}
int operator()() { return 10*(1+(start++ %3)); }; };

int main() {
vector<int> v1(10);
vector<int> v2(10);
generate(v1.begin(), v1.end(), Sequence(1));
sort(v1.rbegin(), v1.rend());
unique_copy(v1.begin(),v1.end(), v2.begin());
for_each(v2.begin(), v2.end(), Out<int>(cout) );cout<<endl;
return 0;
}
```

Program outputs:

- A. 20 30 10 20 30 10 20 30 10 20
- B. 30 20 10 0 0 0 0 0 0 0
- C. 30 0 0 0 0 0 0 0 20 10
- D. compilation error

**Answer: B**

**QUESTION NO: 68**

What happens when you attempt to compile and run the following code?

```
#include <iostream>

#include <algorithm>

#include <set>

using namespace std;

class A {

int a;

public:

A(int a) : a(a) {}

int getA() const { return a; } void setA(int a) { this->a = a; }

operator int() const {return a;}

};

int main () {

int t[] = {1,2,3,2,3,5,1,2,7,3,2,1,10, 4,4,5};

set<A> s (t,t+15);

cout<<equal(s.begin(), s.end(), t)<<endl;

return 0;

}
```

Program outputs:

- A. true
- B. false
- C. 1

- D. 0
- E. compilation error

**Answer: D**

**QUESTION NO: 69**

Which method added to class B at the marked spot will allow the code below to compile? Choose all possible solutions.

```
#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

class B { int val;

public:

B(int v):val(v){}

int getV() const {return val;}

/* Insert Code Here */

};

ostream & operator <<(ostream & out, const B & v) { out<<v.getV(); return out;}

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) { out<<val<<" "; } };

int main() {

int t[]={8, 10, 5, 1, 4, 6, 2, 7, 9, 3};

vector<B> v1(t, t+10);

sort(v1.begin(), v1.end(), greater<B>());
```

```
for_each(v1.begin(), v1.end(), Out<B>(cout));cout<<endl;

return 0;

}
```

- A. bool operator < (const B & v) const { return val<v.val;};
- B. bool operator > (const B & v) const { return val<v.val;};
- C. bool operator > (const B & v) const { return val>v.val;};
- D. bool operator == (const B & v) const { return val==v.val;};
- E. operator int () const { return val; }

**Answer: B,C,D**

### QUESTION NO: 70

What happens when you attempt to compile and run the following code?

```
#include <iostream>

#include <map>

#include <vector>

#include <sstream>

#include <string>

using namespace std;

int main(){

int t[]={ 3, 4, 2, 1, 0, 1, 2, 3, 4, 0 };

vector<int> v(t, t+10);

multimap<int,string> m;

for(vector<int>::iterator i=v.begin(); i!=v.end(); i++) {

stringstream s; s<<*i<<*i; m.insert(pair<int,string>(*i,s.str()));

}

for(multimap<int, string>::iterator i=m.begin();i!= m.end(); i++) {
```



```
cout<<*i<<" ";  
  
}  
  
return 0;  
  
}
```

- A. program outputs: 3 4 2 1 0 1 2 3 4 0
- B. program outputs: 00 11 22 33 44
- C. program outputs: 0 0 1 1 2 2 3 3 4 4
- D. program outputs: 0 0 0 1 1 1 2 2 2 3 3 3 4 4 4
- E. compilation error

**Answer: E**

### QUESTION NO: 71

What happens when you attempt to compile and run the following code?

```
#include <iostream>  
  
#include <algorithm>  
  
#include <deque>  
  
using namespace std;  
  
class A {  
  
int a;  
  
public:  
  
A(int a) : a(a) {}  
  
int getA() const { return a; } void setA(int a) { this->a = a; }  
  
};  
  
int main () {  
  
int t[] = {1,2,3,2,3,5,1,2,7,3,2,1,10, 4,4,5};  
  
deque<int> d (t,t+15);
```

```
int number = count(d.begin(), d.end(), 2);  
  
cout<< number<<endl;  
  
return 0;  
  
}
```

Program outputs:

- A. 4
- B. 3
- C. 2
- D. 0
- E. compilation error

**Answer: A**

### QUESTION NO: 72

What happens when you attempt to compile and run the following code?

```
#include <vector>  
  
#include <iostream>  
  
int main ()  
{  
    std::vector<int>v1;  
    for(int i = 10; i>0; i??)  
    {  
        v1.push_back(i);  
    }  
  
    std::vector<int>::iterator it = v1.begin();  
  
    int sum = 0;  
  
    while(it != v1.end())
```

```
{  
  
sum+=it++;  
  
}  
  
std::cout<<*v1.erase(v1.begin(),v1.end())?3<<" "<<sum <<std::endl;  
  
return 0;  
  
}
```

- A. program outputs 3 55
- B. compilation error
- C. program outputs 3 45
- D. program outputs 7 55

**Answer: B**

### QUESTION NO: 73

What happens when you attempt to compile and run the following code?

```
include <iostream>  
  
#include <algorithm>  
  
#include <vector>  
  
#include <deque>  
  
#include <set>  
  
using namespace std;  
  
int main() {  
  
int t[] = { 10, 5, 9, 6, 2, 4, 7, 8, 3, 1 };  
  
vector<int> v1(t, t + 10);  
  
deque<int> d1(t, t + 10);
```

```
set<int> s1(t, t + 10);

cout<<find(v1.begin(), v1.end(), 6)<<" "<<find(d1.begin(), d1.end(), 6)<<" "<<find(s1.begin(),
s1.end(), 6);

return 0;

}
```

- A. program outputs: 6 6 6
- B. program outputs: 3 3 5
- C. program outputs: 3 6 5
- D. compilation error
- E. none of these

**Answer: D**

#### QUESTION NO: 74

What happens when you attempt to compile and run the following code?

```
#include <iostream>

using namespace std;

template <typename T>

class A {

T _v;

public:

A() {}

A(T v): _v(v){}

T getV() { return _v; }

void add(T a) { _v+=a; }

template <class U>
```

```
U get(U a) {  
    return (U)(_v);  
}  
};  
  
int main()  
{  
    A<int> a(1);  
    a.add(10);  
    cout.setf( ios::showpoint);  
    cout << a.getV() << " " << a.get(1.0)<<endl;  
    return 0;  
}
```

- A. program will display: 11 11
- B. program will not compile
- C. program will display: 11.0000 11
- D. program will display: 11 11.000

**Answer: D**

#### **QUESTION NO: 75**

What happens when you attempt to compile and run the following code?

```
#include <iostream>  
  
using namespace std;  
  
void g(int a)
```

```
{  
cout<<a?1<<endl;  
}
```

```
template<class A>  
void g(A a)  
{  
cout<<a+1<<endl;  
}
```

```
int main()  
{  
int a = 1;  
g(a);  
return 0;  
}
```

- A. program displays: 0
- B. program displays: 2
- C. compilation error
- D. runtime exception

**Answer: A**

#### QUESTION NO: 76

What happens when you attempt to compile and run the following code?

```
#include <vector>
```

```
#include <iostream>

#include <algorithm>

using namespace std;

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) { out<<val<<" "; };

}

int main() {

int t1[]={3,2,4,1,5};

int t2[]={5,6,8,2,1};

vector<int> v1(10);

sort(t1, t1+5);

sort(t2, t2+5);

set_difference(t1,t1+5,t2,t2+5,v1.begin());

for_each(v1.begin(), v1.end(), Out<int>(cout));cout<<endl;

return 0;

}
```

Program outputs:

- A. 1 2 3 4 5 6 8 0 0 0
- B. 3 4 0 0 0 0 0 0 0
- C. 6 8 0 0 0 0 0 0 0
- D. compilation error
- E. 1 2 5 0 0 0 0 0 0

**Answer: B**

**QUESTION NO: 77**

What happens when you attempt to compile and run the following code?

```
#include <list>

#include <iostream>

#include <deque>

using namespace std;

template<class T> void print(T start, T end) {

while (start != end) {

std::cout << *start << " "; start++;

}

}

class A {

int a;

public:

A(int a):a(a){}

operator int () const { return a;}int getA() const { return a;}

};

struct R {

int val;

R(int v):val(v){}

bool operator()(const A & a) { return a>val;};

int main() {

int t1[]={ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

list<A> l1(t1, t1 + 10);

R r(4);l1.remove_if(r);

print(l1.begin(), l1.end()); cout<<endl;
```



```
return 0;
```

```
}
```

- A. program outputs: 1 2 3 4
- B. program outputs: 5 6 7 8 9 10
- C. program outputs: 1 2 3 4 5
- D. program outputs: 6 7 8 9 10

**Answer: A**

### QUESTION NO: 78

What happens when you attempt to compile and run the following code?

```
#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) { out<<val<<" "; };

};

int main() {

int t1[]={3,2,4,1,5};

int t2[]={6,10,8,7,9};

vector<int> v1(10);

sort(t1, t1+5);

sort(t2, t2+5);

merge(t1,t1+5,t2,t2+5,v1.begin());
```

```
for_each(v1.begin(), v1.end(), Out<int>(cout));cout<<endl;

return 0;

}
```

Program outputs:

- A. 1 2 3 4 5 6 10 8 7 9
- B. 3 2 4 1 5 6 7 8 9 10
- C. 3 2 4 1 5 6 10 8 7 9
- D. 1 2 3 4 5 6 7 8 9 10
- E. compilation error

**Answer: D**

#### QUESTION NO: 79

What will happen when you attempt to compile and run the code below, assuming that you enter the following sequence: 1.1 2.2 3.3<enter>?

```
#include <iostream>

#include <string>

using namespace std;

int main ()

{

int a,b,c;

cin>>a>>b>>c;

cout<<a<<b<<c<<endl;

return 0;

}
```

Program will output:

- A. 123
- B. 1 2 3
- C. 1.12.23.3
- D. 1.1 2.2 3.3
- E. none of these

**Answer: E**

### QUESTION NO: 80

What happens when you attempt to compile and run the following code?

```
#include <iostream>

#include <algorithm>

#include <vector>

#include <deque>

#include <set>

using namespace std;

struct display {

void operator() (int i) {cout << " " << i;}

};

int main() {

int t[] = { 10, 5, 9, 6, 2, 4, 7, 8, 3, 1 };

vector<int> v1(t, t + 10);

deque<int> d1(t, t + 10);

set<int> s1(t, t + 10);

for_each(v1.begin(), v1.end(), display); //Line l
```

```
for_each(d1.begin(), d1.end(), *(new display())); // Line II

for_each(s1.begin(), s1.end(), display()); // Line III

return 0;

}
```

- A. program outputs: 10 5 9 6 2 4 7 8 3 1 10 5 9 6 2 4 7 8 3 1 1 2 3 4 5 6 7 8 9 10
- B. program outputs: 10 5 9 6 2 4 7 8 3 1 10 5 9 6 2 4 7 8 3 1 10 5 9 6 2 4 7 8 3 1
- C. compilation error in line I
- D. compilation error in line II
- E. compilation error in line III

**Answer: C**

### QUESTION NO: 81

What happens when you attempt to compile and run the following code?

```
#include <iostream>

#include <algorithm>

#include <vector>

#include <set>

using namespace std;

int main() {

int t[] = {1,2,3,2,3,5,1,2,7,3,2,1,10, 4,4,5};

vector<int> v1(t, t + 15);

set<int> s1(t, t + 15);
```

```
pair<set<int>::iterator, vector<int>::iterator > resultSet = mismatch(s1.begin(), s1.end(),
v1.begin());

cout<<*resultSet.first<<" "<<*resultSet.second<<endl;

return 0;

}
```

Program outputs:

- A. 2 4
- B. 4 2
- C. 0 5
- D. compilation error

**Answer: B**

**QUESTION NO: 82**

Which changes introduced independently will allow the code to compile and display 0 0 1 1 8 8 9 9 (choose all that apply)?

```
#include <iostream>

#include <set>

#include <vector>

using namespace std;

class A {

int a;

public:

A(int a):a(a){}

int getA() const { return a;}
```

```
/* Insert Code Here 1 */  
  
};  
  
/* Insert Code Here 2*/  
  
int main(){  
  
A t[] ={ 3, 4, 2, 1, 6, 5, 7, 9, 8, 0 };  
  
set<A>s(t, t+10);/* Replace Code Here 3 */  
  
multiset<A> s1(s.begin(),s.end());/* Replace Code Here 4 */  
  
s1.insert(s.begin(),s.end());  
  
s1.erase(s1.lower_bound(2),s1.upper_bound(7));  
  
multiset<A>::iterator i=s1.begin();/* Replace Code Here 5 */  
  
for( ;i!= s1.end(); i++)  
  
{  
  
cout<<i?>getA()<<" ";  
  
}  
  
cout<<endl;  
  
return 0;  
  
}
```

- A. operator int() const { return a;} inserted at Place 1
- B. bool operator < (const A & b) const { return a<b.a;} inserted at Place 1
- C. bool operator < (const A & b) const { return b.a<a;} inserted at Place 1
- D. struct R { bool operator ()(const A & a, const A & b) { return a.getA()<b.getA();} }; inserted at Place 2

replacing line marked 3 with set<A, R>s(t, t+10);

replacing line marked 4 with multiset<A,R> s1(s.begin(),s.end());

replacing line marked 5 with multiset<A,R>::iterator i=s1.begin();

**Answer: A,B,D**

**QUESTION NO: 83**

What happens when you attempt to compile and run the following code?

```
#include <deque>

#include <iostream>

#include <algorithm>

using namespace std;

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) { out<<val<<" "; } };

int main() {

int t[]={8, 10, 5, 1, 4, 6, 2, 7, 9, 3};

deque<int> d1(t, t+10);

deque<int>::iterator it = lower_bound(d1.begin(), d1.end(), 4);

for_each(it, d1.end(), Out<int>(cout));cout<<endl;

return 0;

}
```

Program outputs:

- A. 8 10 5 1 4 6 2 7 9 3
- B. 4 5 6 7 8 9 10
- C. 1 2 3 4 5 6 7 8 9 10
- D. compilation error
- E. 1 2 3 4

**Answer: A**

**QUESTION NO: 84**

What happens when you attempt to compile and run the following code? Choose all that apply.

```
#include <iostream>

#include <fstream>

#include <string>

#include <list>

#include <algorithm>

#include <iomanip>

using namespace std;

class B { int val;

public:

B(int v=0):val(v){}

int getV() const {return val;}

operator int() const { return val; };};

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) {out<<setw(3)<<hex<<val; } };

int main () {

int t[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

fstream f("test.out", ios::trunc|ios::out);

list<B> l(t, t+10);

for_each(l.begin(), l.end(), Out<B>(f));

f.close();

f.open("test.out");
```



```
for( ; f.good() ; ) {  
    int i;  
    f>>i;  
    cout<<i<<" ";  
}  
f.close();  
return 0;  
}
```

- A. file test.out will be opened writing
- B. file test.out will be truncated
- C. file test.out will be opened for reading
- D. no file will be created nor opened
- E. program will display sequence 1 2 3 4 5 6 7 8 9 10

**Answer: A,B,C,E**

### QUESTION NO: 85

What happens when you attempt to compile and run the following code?

```
#include <vector>  
#include <iostream>  
#include <algorithm>  
using namespace std;  
class B { int val;  
public:  
    B(int v):val(v){}  
    int getV() const {return val;}  bool operator > (const B & v) const { return val>v.val;} };  
ostream & operator <<(ostream & out, const B & v) { out<<v.getV(); return out;}
```

```
template<class T>struct Out {  
  
    ostream & out;  
  
    Out(ostream & o): out(o){}  
  
    void operator() (const T & val ) { out<<val<<" "; } };  
  
int main() {  
  
    B t[]={3,2,4,1,5,10,9,7,8,6};  
  
    vector<B> v1(t,t+10);  
  
    sort(v1.begin(), v1.end(), greater<B>());  
  
    cout<<*min_element(v1.begin(), v1.end());  
  
    return 0;  
  
}
```

Program outputs:

- A. 3
- B. 1
- C. 6
- D. 10
- E. compilation error

**Answer: E**

### QUESTION NO: 86

What will happen when you attempt to compile and run the code below, assuming that you enter the following sequence: 64 100<enter>?

```
#include <iostream>  
  
#include <string>  
  
#include <sstream>
```

```
#include <iomanip>

using namespace std;

int main ()
{
string s;
getline(cin, s);
stringstream input(s);
stringstream output;

for( ; !input.fail() ; )
{
int i;
input>>hex>>i;
output<<setw(4)<<i;
}
cout<<output.str();
return 0;
}
```

What will be the result assuming that user will enter following sequence: 64 100:

- A. 64 100
- B. 100 256
- C. 100 256 256
- D. 0x64 0x100
- E. 0x100 0x256 0x256

**Answer: C**

**QUESTION NO: 87**

What happens when you attempt to compile and run the following code?

```
#include <vector>

#include <iostream>

#include <algorithm>

#include <functional>

using namespace std;

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) { out<<val<<" "; } };

int Add(int a, int b) {

return a+b;

}

int main() {

int t[]={1,2,3,4,5,6,7,8,9,10};

vector<int> v1(t, t+10);

vector<int> v2(10);

transform(v1.begin(), v1.end(), v2.begin(), bind2nd(ptr_fun (Add),1));

for_each(v2.rbegin(), v2.rend(), Out<int>(cout));cout<<endl;

return 0;

}
```

Program outputs:

- A. 1 2 3 4 5 6 7 8 9 10
- B. 2 3 4 5 6 7 8 9 10 11
- C. 10 9 8 7 6 5 4 3 2 1
- D. 11 10 9 8 7 6 5 4 3 2
- E. compilation error

**Answer: D**

### QUESTION NO: 88

What happens when you attempt to compile and run the following code?

```
#include <vector>

#include <set>

#include <iostream>

#include <algorithm>

using namespace std;

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) { out<<val<<" "; } };

struct Sequence {

int start;

Sequence(int start):start(start){}

int operator()() { return start++; } };

int main() {

vector<int> v1(10);
```

```
generate_n(v1.begin(), 10, Sequence(1));  
random_shuffle(v1.rbegin(), v1.rend());  
sort(v1.begin(), v1.end(), great<int>());  
for_each(v1.begin(), v1.end(), Out<int>(cout));cout<<endl;  
return 0;  
}
```

Program outputs:

- A. 8 10 5 1 4 6 2 7 9 3
- B. 1 2 3 4 5 6 7 8 9 10
- C. compilation error
- D. 10 9 8 7 6 5 4 3 2 1

**Answer: C**

#### QUESTION NO: 89

What happens when you attempt to compile and run the following code?

```
#include <iostream>  
  
#include <algorithm>  
  
#include <map>  
  
using namespace std;  
  
void myfunction(pair<int, int> i) {  
    cout << " " << i.first;  
}  
  
int main() {  
    int t[] = { 10, 5, 9, 6, 2, 4, 7, 8, 3, 1 };  
}
```

```
map<int, int> m;

for(int i=0; i < 10; i++) {

    m[i]=t[i];

}

for_each(m.begin(), m.end(), myfunction);

return 0;

}
```

Program outputs:

- A. 10 5 9 6 2 4 7 8 3 1
- B. 0 1 2 3 4 5 6 7 8 9
- C. 9 8 7 6 5 4 3 2 1 0
- D. 1 3 8 7 4 2 6 9 5 10
- E. compilation error

**Answer: B**

### QUESTION NO: 90

What happens when you attempt to compile and run the following code?

```
#include <iostream>

using namespace std;

int main()

{

cout<<true<<" "<<boolalpha<<false;

return 0;

}
```

Program outputs:

- A. true false
- B. 1 0
- C. 1 false
- D. true 0
- E. compilation error

**Answer: C**

### QUESTION NO: 91

What happens when you attempt to compile and run the following code?

```
#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

class B { int val;

public:

B(int v):val(v){}

int getV() const {return val;} bool operator < (const B & v) const { return val<v.val;} };

ostream & operator <<(ostream & out, const B & v) { out<<v.getV(); return out;}

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) { out<<val<<" "; } };

int main() {

int t[]={8, 10, 5, 1, 4, 6, 2, 7, 9, 3};

vector<B> v1(t, t+10);

sort(v1.begin(), v1.end());

for_each(v1.begin(), v1.end(), Out<B>(cout));cout<<endl;
```



```
return 0;
```

```
}
```

Program outputs:

**A.** 8 10 5 1 4 6 2 7 9 3

**B.** 1 2 3 4 5 6 7 8 9 10

**C.** compilation error

**D.** 10 9 8 7 6 5 4 3 2 1

**Answer: B**

### QUESTION NO: 92

Which sentence is correct about the code below? Choose all that apply.

```
#include <iostream>
```

```
#include <algorithm>
```

```
#include <vector>
```

```
using namespace std;
```

```
class F {
```

```
int val;
```

```
public:
```

```
F(int v):val(v){}
```

```
bool operator() (int v) {
```

```
if (v == val) return true;
```

```
return false;
```

```
}
```

```
};
```

```
int main() {
```

```
int t[] = { 10, 5, 9, 6, 2, 4, 7, 8, 3, 1 };  
  
vector<int> v1(t, t + 10);  
  
if (find(v1.begin(), v1.end(), 6) == find(v1.begin(), v1.end(), F(6))) {  
    cout<<"Found!\n";  
  
} else {  
  
    cout<<"Not found!\n";  
  
}  
  
return 0;  
  
}
```

- A. it will compile successfully
- B. it will display Found!
- C. it will display Not found!
- D. it will not compile successfully

**Answer: D**

### QUESTION NO: 93

What will happen when you attempt to compile and run the following code?

```
#include <deque>  
  
#include <vector>  
  
#include <iostream>  
  
using namespace std;  
  
int main ()  
{  
  
    vector<int>v1;  
  
    deque<int>d1;  
  
    for(int i=0; i<5; i++)
```

```
{  
v1.push_back(i);v1.push_front(i);  
d1.push_back(i);d1.push_front(i);  
}  
for(int i=0; i<d1.size(); i++)  
{  
cout<<d1[i]<<" "<<v1[i]<<" ";  
}  
cout<<endl;  
return 0;  
}
```

What will be its output:

- A. 4 4 3 3 2 2 1 1 0 0 0 0 1 1 2 2 3 3 4 4
- B. runtime exception
- C. compilation error due to line 11
- D. compilation error due to line 12

**Answer: C**

#### QUESTION NO: 94

What happens when you attempt to compile and run the following code?

```
#include <iostream>  
  
#include <algorithm>  
  
#include <deque>  
  
#include <vector>  
  
using namespace std;  
  
bool identical(int a, int b) {
```

```
return b == 2*a?true:false;

}

int main() {

int t[] = {1,2,3,2,3,5,1,2,7,3,2,1,10, 4,4,5};

int u[] = {2,4,6,4,6,10,2,4,14,6,4,2,20,8,8,5};

vector<int> v1(t, t + 15);

deque<int> d1(u, u + 15);

pair<deque<int>::iterator, vector<int>::iterator > result;

result = mismatch(d1.begin(), d1.end(), v1.begin(), identical); //Line I

if (result.first == d1.end() && result.second == v1.end()) { //Line II

cout<<"Identical\n";

} else {

cout<<"Not identical\n";

}

return 0;

}
```

Program outputs:

- A. Identical
- B. Not identical
- C. compilation error at line marked I
- D. compilation error at line marked II

**Answer: B**

#### QUESTION NO: 95

What will happen when you attempt to compile and run the code below, assuming that you enter

the following sequence: true false<enter>?

```
#include <iostream>

#include <string>

using namespace std;

int main ()
{
bool a,b;

cin>>boolalpha>>a>>b;

cout<<a<<b<<endl;

return 0;
}
```

Program will output:

- A. truefalse
- B. true0;
- C. 1false
- D. 10
- E. none of these

**Answer: D**

#### **QUESTION NO: 96**

What happens when you attempt to compile and run the following code?

```
#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;
```

```
class B { int val;

public:

B(int v):val(v){}

int getV() const {return val;}  bool operator < (const B & v) const { return val<v.val;} };

ostream & operator <<(ostream & out, const B & v) { out<<v.getV(); return out;}

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) { out<<val<<" "; } };

int main() {

int t[]={8, 10, 5, 1, 4, 6, 2, 7, 9, 3};

vector<B> v1(t, t+10);

sort(v1.begin(), v1.end(), greater<B>());

for_each(v1.begin(), v1.end(), Out<B>(cout));cout<<endl;

return 0;

}
```

Program outputs:

- A. 8 10 5 1 4 6 2 7 9 3
- B. 1 2 3 4 5 6 7 8 9 10
- C. compilation error
- D. 10 9 8 7 6 5 4 3 2 1

**Answer: C**

#### QUESTION NO: 97

What happens when you attempt to compile and run the following code?

```
#include <iostream>
```

```
using namespace std;
```

```
template <class T>
```

```
class A {
```

```
    T _v;
```

```
public:
```

```
    A() {}
```

```
    A(T v): _v(v){}
```

```
friend ostream & operator<<(ostream & c, const A<T> & v);
```

```
};
```

```
template <class T>
```

```
ostream & operator<<(ostream & c, const A<T> & v) {
```

```
    c<<v._v;return c; }
```

```
int main()
```

```
{
```

```
    A<int>a(10);
```

```
    cout<<a<<endl;
```

```
    return 0;
```

```
}
```

- A. program will display:10
- B. program will not compile
- C. program will compile
- D. program will run without output

**Answer: B**

**QUESTION NO: 98**

What happens when you attempt to compile and run the following code?

```
#include <vector>

#include <iostream>

#include <algorithm>

#include <functional>

using namespace std;

class B { int val;

public:

B(int v=0):val(v){}

int getV() const {return val;}

operator int () const { return val; } };

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) { out<<val<<" "; } };

int main() {

B t[]={3,2,4,1,5,6,10,8,7,9};

vector<B> v1(t, t+10);

for_each(v1.begin(), v1.end(), bind1st(plus<B>(), 1));

for_each(v1.rbegin(), v1.rend(), Out<B>(cout));cout<<endl;

return 0;
```



```
}
```

Program outputs:

- A. 3 2 4 1 5 6 10 8 7 9
- B. 4 3 5 2 6 7 11 9 8 10
- C. 9 7 8 10 6 5 1 4 2 3
- D. 10 8 9 11 7 6 2 5 3 4
- E. compilation error

**Answer: C**

### QUESTION NO: 99

What happens when you attempt to compile and run the following code?

```
#include <iostream>

#include <algorithm>

#include <vector>

using namespace std;

class A {

int a;

public:

A(int a) : a(a) {}

int getA() const { return a; } void setA(int a) { this->a = a; }

bool operator==(A & b) { return a == b.a; }

};

struct Compare{

bool operator()(const A & a, const A & b) {return a.getA()==b.getA();}

};

int main () {
```

```
int t[] = {1,2,3,4,5,1,2,3,4,5};  
  
vector<A> v (t,t+10);  
  
vector<A>::iterator it;  
  
A m1[] = {A(1), A(2), A(3)};  
  
it = search (v.begin(), v.end(), m1, m1+3, Compare());  
  
cout << "First found at position: " << it?v.begin() << endl;  
  
return 0;  
  
}
```

Program outputs:

- A. First found at position: 5
- B. First found at position: 0
- C. First found at position: 7
- D. compilation error
- E. First found at position: 10

**Answer: B**

### QUESTION NO: 100

What happens when you attempt to compile and run the following code?

```
#include <deque>  
  
#include <iostream>  
  
#include <algorithm>  
  
using namespace std;  
  
class B { int val;  
  
public:  
  
B(int v):val(v){} B(){}  
  
int getV() const {return val;} };
```

```
ostream & operator <<(ostream & out, const B & v) { out<<v.getV(); return out;}
```

```
template<class T>struct Out {
```

```
    ostream & out;
```

```
    Out(ostream & o): out(o){}
```

```
    void operator() (const T & val ) { out<<val<<" "; } };
```

```
int main() {
```

```
    int t[]={8, 10, 5, 1, 4, 6, 2, 7, 9, 3};
```

```
    deque<B> d1(t, t+10);
```

```
    deque<B>::iterator it = lower_bound(d1.begin(), d1.end(), 4);
```

```
    for_each(it, d1.end(), Out<B>(cout));cout<<endl;
```

```
    return 0;
```

```
}
```

Program outputs:

**A.** 8 10 5 1 4 6 2 7 9 3

**B.** 4 5 6 7 8 9 10

**C.** 1 2 3 4 5 6 7 8 9 10

**D.** compilation error

**E.** 1 2 3 4

**Answer: D**

### QUESTION NO: 101

What happens when you attempt to compile and run the following code?

```
#include <deque>
```

```
#include <iostream>
```

```
#include <algorithm>
```

```
using namespace std;
```

```
class B { int val;

public:

B(int v):val(v){}

int getV() const {return val;}  bool operator < (const B & v) const { return val<v.val;} };

ostream & operator <<(ostream & out, const B & v) { out<<v.getV(); return out;}

template<class T>struct Out {

ostream & out;  Out(ostream & o): out(o){}

void operator() (const T & val ) { out<<val<<" "; } };

int main() {

int t[]={20, 30, 10, 20, 30, 10, 20, 30, 10, 20};

deque<B> d1(t, t+10);

sort(d1.begin(), d1.end());

pair<deque<B> ::iterator, deque<B>::iterator > result = equal_range(d1.begin(), d1.end(), B(20));

for_each(result.first, result.second, Out<B>(cout));cout<<endl;

return 0;

}
```

Program outputs:

- A. 10 10 10 20 20 20 20 30 30 30
- B. 20 20 20 20
- C. 10 20 20 20 20
- D. 20 20 20 20 30
- E. 10 20 20 20 20 30

**Answer: B**

## QUESTION NO: 102

What happens when you attempt to compile and run the following code?

```
#include <iostream>

#include <deque>

#include <list>

#include <queue>

#include <vector>

using namespace std;

int main()

{

int t[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

deque<int> mydeck(t, t+10);list<int> mylist(t,t+10);

queue<int> first;

queue<int> second(mydeck);

queue<int> third(second);

queue<int, list<int> > fourth(mylist);

mylist.clear();third.clear();

cout<<third.size()<< " " <<mydeck.size()<< endl;

cout<<fourth.size()<< " " <<mylist.size()<<endl;

return 0;

}
```

- A. program outputs: 10 0  
10 0
- B. program outputs: 0 0  
0 0
- C. program outputs: 10 10  
10 10
- D. program outputs: 10 0  
0 10
- E. compilation error

**Answer: E**

**QUESTION NO: 103**

What happens when you attempt to compile and run the following code?

```
#include <iostream>

#include <deque>

#include <list>

#include <queue>

#include <vector>

using namespace std;

class compare {

bool reverse;

public:

compare(bool revparam = false){ reverse = revparam;}

bool operator()(int lhs, int rhs) const{

if (reverse)return (lhs > rhs);

elsereturn (lhs < rhs);

}

};

int main(){

int myints[] ={ 3, 4, 2, 1, 6, 5, 7, 9, 8, 0 };

priority_queue<int, deque<int> > first(myints, myints + 10);

priority_queue<int, vector<int>, compare> second(myints, myints + 10,

compare(false));

while (first.size() > 0){
```

```
cout << first.top() << " "; first.pop();  
  
}  
  
while (second.size() > 0) {  
    cout << second.top() << " ";second.pop();  
}  
  
return 0;  
  
}
```

- A. compilation error
- B. program outputs: 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
- C. program outputs: 9 8 7 6 5 4 3 2 1 0 0 1 2 3 4 5 6 7 8 9
- D. program outputs: 3 4 2 1 6 5 7 9 8 0 3 4 2 1 6 5 7 9 8 0

**Answer: B**

#### QUESTION NO: 104

What happens when you attempt to compile and run the following code?

```
#include <deque>  
  
#include <iostream>  
  
#include <algorithm>  
  
#include <set>  
  
using namespace std;  
  
template<class T>struct Out {  
    ostream & out;  
    Out(ostream & o): out(o){}  
    void operator() (const T & val ) { out<<val<<" "; } };  
  
int main() {  
    int t[]={8, 10, 5, 1, 4, 6, 2, 7, 9, 3};
```

```
deque<int> d1(t, t+10);  
  
set<int> s1(t,t+10);  
  
cout<<binary_search(s1.begin(),s1.end(), 4)<<" "<<binary_search(d1.begin(),d1.end(), 4)<<endl;  
  
return 0;  
  
}
```

Choose all possible outputs (all that apply):

- A. 1 0
- B. 1 1
- C. true true
- D. false false
- E. compilation error

**Answer: A,B**

### QUESTION NO: 105

What happens when you attempt to compile and run the following code?

```
#include <vector>  
  
#include <iostream>  
  
#include <algorithm>  
  
using namespace std;  
  
template<class T>struct Out {  
  
    ostream & out;  
  
    Out(ostream & o): out(o){}  
  
    void operator() (const T & val ) { out<<val<<" "; }  
  
};  
  
struct Add {  
  
    int operator()(int & a, int & b) {
```



```
return a+b;

}

};

int main() {

int t[]={1,2,3,4,5,6,7,8,9,10};

vector<int> v1(t, t+10);

vector<int> v2(10);

transform(v1.begin(), v1.end(), v2.begin(), bind2nd(Add(),1));

for_each(v2.rbegin(), v2.rend(), Out<int>(cout));cout<<endl;

return 0;

}
```

Program outputs:

- A. 1 2 3 4 5 6 7 8 9 10
- B. 2 3 4 5 6 7 8 9 10 11
- C. 10 9 8 7 6 5 4 3 2 1
- D. 11 10 9 8 7 6 5 4 3 2
- E. compilation error

**Answer: E**

### QUESTION NO: 106

What will happen when you attempt to compile and run the code below, assuming you enter the following sequence: 1 2 3<enter>?

```
#include <iostream>
```

```
using namespace std;
```

```
int main ()  
  
{  
  
int a,b,c;  
  
cin>>a>>b>>c;  
  
cout<<a<<b<<c<<endl;  
  
return 0;  
  
}
```

Program will output:

- A. 123
- B. 1 2 3
- C. 321
- D. compilation error
- E. the result is unspecified

**Answer: A**

#### QUESTION NO: 107

What happens when you attempt to compile and run the following code?

```
#include <iostream>  
  
#include <map>  
  
#include <vector>  
  
#include <string>  
  
using namespace std;  
  
int main(){  
  
int second[] = { 3, 4, 2, 1, 6, 5, 7, 9, 8, 0 };  
  
string first[] = {"three", "four", "two", "one", "six", "five", "seven", "nine", "eight", "zero"};  
  
map<int,string> m;
```

```
for(int i=0; i<10; i++) {  
    m.insert(pair<int,string>(second[i],first[i]));  
}  
m[0]="ten";  
m.insert(pair<int,string>(1,"eleven"));  
for(map<int, string>::iterator i=m.begin();i!= m.end(); i++) {  
    cout<<i?>second<<" ";  
}  
return 0;  
}
```

- A. program outputs: zero one two three four five six seven eight nine
- B. program outputs: ten one two three four five six seven eight nine
- C. program outputs: zero eleven two three four five six seven eight nine
- D. program outputs: ten eleven two three four five six seven eight nine
- E. program outputs: 0 1 2 3 4 5 6 7 8 9

**Answer: B**

### QUESTION NO: 108

What will happen when you attempt to compile and run the code below, assuming that you enter the following sequence: 1 2 3<enter>?

```
#include <iostream>  
  
#include <string>  
  
#include <sstream>  
  
using namespace std;  
  
int main ()  
{
```

```
string s;

getline(cin, s);

stringstream input(s);

stringstream output;

for( ; !input.fail() ; )

{

    int i;

    input>>i;

    output<<hex<<i<<" ";

}

cout<<output.str();

return 0;

}
```

Program will output:

- A. 1 2 3
- B. 1 2 3 3
- C. 0x1 0x2 0x3
- D. 0x1 0x2 0x3 0x3
- E. program runs forever without output

**Answer: B**

### QUESTION NO: 109

What happens when you attempt to compile and run the following code?

```
#include <iostream>

#include <algorithm>
```

```
#include <vector>

using namespace std;

struct Compare {
bool operator()(int a) {
    if (a >5) return true;

    return false;
}
};

int main () {
int t[] = {1,2,3,2,3,5,1,2,7,3,2,1,10, 4,4,5};

vector<int> v (t,t+15);

int number = count(v.begin(), v.end(), Compare());

cout<< number<<endl;

return 0;
}
```

Program outputs:

- A. 4
- B. 3
- C. 2
- D. 0
- E. compilation error

**Answer: E**

#### **QUESTION NO: 110**

What happens when you attempt to compile and run the following code?

```
#include <deque>

#include <set>

#include <iostream>

#include <algorithm>

using namespace std;

class B { int val;

public:

B(int v):val(v){}

int getV() const {return val;} bool operator < (const B & v) const { return val<v.val;} };

ostream & operator <<(ostream & out, const B & v) { out<<v.getV(); return out;}

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) { out<<val<<" "; } };

int main() {

int t[]={8, 10, 5, 1, 4, 6, 2, 7, 9, 3};

deque<B> d1(t, t+10);

sort(d1.begin(), d1.end());

set<B> s1(t,t+10);

cout<<binary_search(s1.begin(),s1.end(), 4)<<" " <<binary_search(d1.begin(),d1.end(), 4)<<endl;

return 0;

}
```

Program outputs:

- A. 1 0
- B. 1 1
- C. true true
- D. false false
- E. compilation error

**Answer: E**

## Topic 2, Volume B

### QUESTION NO: 111

What happens when you attempt to compile and run the following code?

```
#include <vector>

#include <iostream>

using namespace std;

class A
{
int a,b;

public:

A(const A & c) { a = c.a; }

A():a(0),b(0){}

void setA(int a) {this->a = a;} void setB(int b) {this->b = b;}

int getA() {return a;} int getB() {return b;}

};

int main ()
{
vector<A>v;

A a;

a.setA(10); a.setB(11);

v.push_back(a);

cout<<v[0].getB()<<" "<<v[0].getA()<<endl;
```

```
return 0;
```

```
}
```

- A. program outputs 10 11
- B. the result is unpredictable
- C. program outputs 10 0
- D. program outputs 11 0
- E. compilation error

**Answer: B**

### QUESTION NO: 112

What happens when you attempt to compile and run the following code?

```
#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

class B { int val;

public:

B(int v=0):val(v){}

int getV() const {return val;}

operator int () const { return val; } };

template<class T>struct Out {

ostream & out;

Out(ostream & o): out(o){}

void operator() (const T & val ) { out<<val<<" "; } };
```



```
struct Add {  
    B operator()(B & a, B & b) { return a+b; };  
};  
  
int main() {  
    int t[]={1,2,3,4,5,6,7,8,9,10};  
    vector<B> v1(t, t+10);  
    vector<B> v2(10);  
    transform(v1.begin(), v1.end(), v2.begin(), bind1st(Add(),1));  
    for_each(v2.rbegin(), v2.rend(), Out<B>(cout));cout<<endl;  
    return 0;  
}
```

Program outputs:

- A. 1 2 3 4 5 6 7 8 9 10
- B. 2 3 4 5 6 7 8 9 10 11
- C. 10 9 8 7 6 5 4 3 2 1
- D. 11 10 9 8 7 6 5 4 3 2
- E. compilation error

**Answer: E**

### QUESTION NO: 113

What happens when you attempt to compile and run the following code?

```
#include <vector>  
  
#include <iostream>  
  
#include <algorithm>  
  
using namespace std;  
  
template<class T>struct Out {  
    ostream & out;
```

```
Out(ostream & o): out(o){}
```

```
void operator() (const T & val ) { out<<val<<" "; } };
```

```
int main() {  
int t1[]={3,2,4,1,5};  
int t2[]={5,6,8,2,1};  
vector<int> v1(10);  
sort(t1, t1+5);  
sort(t2, t2+5);  
set_symmetric_difference(t1,t1+5,t2,t2+5,v1.begin());  
for_each(v1.begin(), v1.end(), Out<int>(cout));cout<<endl;  
return 0;  
}
```

Program outputs:

- A. 6 8 3 4 0 0 0 0 0 0
- B. 3 4 0 0 0 0 0 0 0 0
- C. 6 8 0 0 0 0 0 0 0 0
- D. compilation error
- E. 3 4 6 8 0 0 0 0 0 0

**Answer: E**

To Read the [Whole Q&As](#), please purchase the [Complete Version](#) from [Our website](#).

# Trying our product !

- ★ **100%** Guaranteed Success
- ★ **100%** Money Back Guarantee
- ★ **365 Days** Free Update
- ★ **Instant Download** After Purchase
- ★ **24x7** Customer Support
- ★ Average **99.9%** Success Rate
- ★ More than **69,000** Satisfied Customers Worldwide
- ★ Multi-Platform capabilities - **Windows, Mac, Android, iPhone, iPod, iPad, Kindle**

## Need Help

Please provide as much detail as possible so we can best assist you.

To update a previously submitted ticket:



 <b>One Year Free Update</b> <p>Free update is available within One Year after your purchase. After One Year, you will get 50% discounts for updating. And we are proud to boast a 24/7 efficient Customer Support system via Email.</p>	 <b>Money Back Guarantee</b> <p>To ensure that you are spending on quality products, we provide 100% money back guarantee for 30 days from the date of purchase.</p>	 <b>Security &amp; Privacy</b> <p>We respect customer privacy. We use McAfee's security service to provide you with utmost security for your personal information &amp; peace of mind.</p>
---	---	--

[Guarantee & Policy](#) | [Privacy & Policy](#) | [Terms & Conditions](#)

Any charges made through this site will appear as Global Simulators Limited.

All trademarks are the property of their respective owners.

Copyright © 2004-2015, All Rights Reserved.