

## Question: 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: 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: 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

## Question: 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: 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: 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 0
- E. 1 2 5 0 0 0 0 0 0 0

**Answer: D**

## Question: 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: 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

**Question: 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: 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: 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: 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

**Answer: E**

## Question: 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: 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: 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**

