1. If you enter 1 0, what is the output of the following code?

#include <iostream>

using namespace std;

int main()

{

int number1, number2;

cin >> number1 >> number2;

try

{

if (number2 == 0)

throw number1;

cout << number1 / number2 << endl;

cout << "C";

}

catch (int e)

{

cout << "A";

}

cout << "B";

return 0;

}

1. A
2. B
3. C
4. **AB**
5. Predict the output

#include<iostream>

using namespace std;

int main()

{

cout.width(10);

cout.fill('\*');

cout <<"Done";

}

1. DONE
2. DONE\*\*\*\*\*\*
3. **\*\*\*\*\*\*DONE**
4. \*DONE
5. Choose the appropriate option

Which of the following classes are in the header file

<stdexcept>?

1. runtime\_error
2. overflow\_error
3. underflow\_error
4. **All of the above**
5. **Predict the output:**

#include <iostream>

int main () {

int p = 1;

int z = 0;

int n = -1;

cout << showpos << p << " " << z << " " << n ;

cout << noshowpos << p << " " << z << " " << n ;

return 0;

}

1. 1 0 -1 +1 +0 -1
2. +1 +0 -1 +1 +0 -1
3. **+1 +0 -1 1 0 1**
4. 1 0 -1 +1 0 -1
5. To add an int value 5 to a vector v of integers, use
6. v.add(5);
7. v.insert(5);
8. v.push\_back(5);
9. v.append(5);
10. **Predict the output:**

#include <iostream>

#include <deque>

using namespace std;

int main ()

{

deque<int> mydeque;

mydeque.push\_back(100);

mydeque.push\_front(100);

mydeque.push\_back(200);

mydeque.push\_back(300);

mydeque.push\_front(100);

mydeque.push\_front(200);

mydeque.push\_front(300);

int sum=0;

while (!mydeque.empty())

{

sum+=mydeque.back();

mydeque.pop\_back();

}

cout << sum <<endl;

return 0;

}

1. 1000
2. **1300**
3. 1100
4. 1200
5. **Predict the output:**

#include <iostream>

#include <set>

#include <iterator>

using namespace std;

int main()

{

// empty set container

multiset <int, greater <int> > g;

// insert elements in random order

g.insert(40);

g.insert(30);

g.insert(60);

g.insert(20);

g.insert(50);

g.insert(50); // only one 50 will be added to the set

g.insert(10);

// printing set g

set <int, greater <int> > :: iterator itr;

for (itr = g.begin(); itr != g.end(); ++itr)

{

cout << \*itr<<" ";

}

cout << endl;

return 0;

}

1. **60 50 50 40 30 20 10**
2. 60 50 40 30 20 10
3. 10 20 30 40 50 60
4. 10 20 30 40 50 50 60
5. **Choose appropriate option:**

#include <iostream>

#include <vector>

#include <string>

using namespace std;

int main()

{

vector<int> v1(5);

for (int i = 0; i < v1.size(); ++i) {

v1[i] = (i+1) \* 2;

}

for (int i = 0; i < v1.size(); ++i) {

cout << v1.at(i) << " ";

}

cout << endl;

return 0;

}

1. 1 2 3 4 5
2. **2 4 6 8 10**
3. 10 8 6 4 2
4. 3 5 7 9 11
5. **What is output of following code?**

#include <iostream>

#include <algorithm>

#include <vector>

using namespace std;

int main ()

{

int myints[]={ 10, 20, 30, 40, 50 };

vector<int> myvector (4, 99);

iter\_swap(myints, myvector.begin());

iter\_swap(myints + 3,myvector.begin() + 2);

for (vector<int> :: iterator it = myvector.begin();

it != myvector.end(); ++it)

cout << ' ' << \*it;

return 0;

}

1. 10 40
2. **10 99 40 99**
3. 10 20 99 40 99
4. 99 20 30 99
5. What is output of following code?

#include <iostream>

#include <vector>

#include<iterator>

using namespace std;

int main ()

{

vector<int> myvector;

for (int i = 1; i <= 10; i++)

myvector.push\_back(i);

myvector.erase (myvector.begin() + 6);

myvector.erase (myvector.begin(),

myvector.begin() + 4);

for (unsigned i = 0; i < myvector.size(); ++i)

cout << ' ' << myvector[i];

return 0;

}

1. 5 6 7 8 9
2. **5 6 8 9 10**
3. 6 7 8 9 10
4. 4 5 6 8 9 10
5. What is the output of following code?

#include <iostream>

#include <iterator>

#include <list>

using namespace std;

int main ()

{

list<int> firstlist, secondlist;

for (int i = 1; i <= 2; i++)

{

firstlist.push\_back(i);

secondlist.push\_back(i \* 10);

}

list<int> :: iterator it;

it = firstlist.begin();

advance (it, 3);

copy (secondlist.begin(), secondlist.end(),

inserter(firstlist, it));

for ( it = firstlist.begin(); it != firstlist.end(); ++it )

cout << \*it << " ";

return 0;

}

1. **10 20 1 2**
2. 10 20
3. 1 2
4. 1 10
5. What will be output?

int main()

{

list <int> ilist;

ilist.push\_back(10);

ilist.push\_back(20);

ilist.push\_front(30);

ilist.push\_front(40);

ilist.reverse();

int s = ilist.size();

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

{c

out<<ilist.front()<<endl;

ilist.pop\_front();

}}

1. 40 30 10 20
2. **20 10 30 40**
3. 30 10 20 40
4. 40 30 20 10
5. Predict the output

#include <iostream>

#include <deque>

using namespace std;

int main ()

{

deque<int> mydeque (5); // 5 default-constructed ints

deque<int>::reverse\_iterator rit = mydeque.rbegin();

int i=0;

for (rit = mydeque.rbegin(); rit!= mydeque.rend();

++rit)

\*rit = ++i;

for (deque<int>::iterator it = mydeque.begin(); it !=

mydeque.end(); ++it)

cout << ' ' << \*it;

return 0;

}

1. 1 2 3 4 5
2. **5 4 3 2 1**
3. 0 1 2 3 4
4. 4 3 2 1 0
5. Predict the output

#include <iostream>

#include <set>

#include <iterator>

using namespace std;

int main()

{

set <int, greater <int> > g;

g.insert(40);

g.insert(30);

g.insert(60);

g.insert(20);

g.insert(50);

g.insert(50);

g.insert(10);

set <int, greater <int> > :: iterator itr;

cout << "\nThe set gquiz1 is : ";

for (set <int, greater <int> > :: iterator itr = g.begin();

itr != g.end(); ++itr)

{

cout << " " << \*itr;

}

cout << endl;

return 0;

}

1. **60 50 40 30 20 10**
2. 10 20 30 40 50 60
3. 10 20 30 40 50 50 60
4. 60 50 50 40 30 20 10