Fall Semester 15, Dr. Punch. Exam #2 (11/12), form 2 A

Last name (printed):____

First name (printed):____

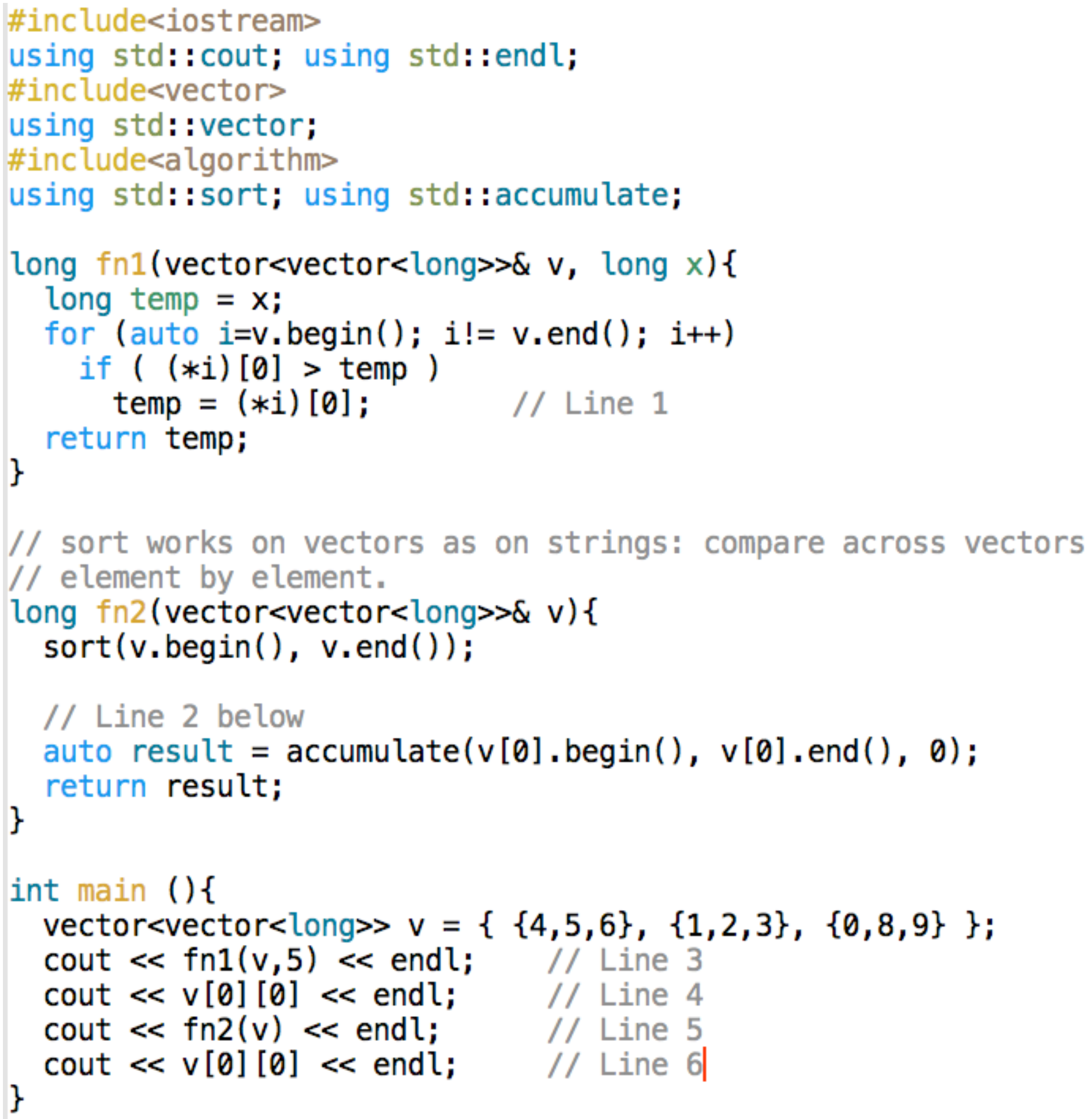
Directions:

a) DO NOT OPEN YOUR EXAM BOOKLET UNTIL YOU HAVE BEEN TOLD TO BEGIN.
b) You have 80 minutes to complete the exam (10:20-11:40)
c) This exam booklet contains 30 multiple choice questions, each weighted equally (5 points). **5, double-sided, pages total**
d) You may use one 8.5" x 11" note sheet during the exam. No other reference materials or calculating devices may be used during the examination.
e) Questions will not be interpreted during the examination.
f) You should choose the single best alternative for each question, even if you believe that a question is ambiguous or contains a typographic error.
g) Please fill in the requested information at the top of this exam booklet.
h) Use a #2 pencil to encode any information on the OMR form.
i) Please encode the following on the OMR form:
   - Last name and first initial
   - MSU PID
   - Exam form (see the title of this page)

j) Please sign the OMR form.
k) Only answers recorded on your OMR form will be counted for credit.
l) Completely erase any responses on the OMR form that you wish to delete.
m) You must turn in this exam booklet and the OMR form when you have completed the exam. When leaving, please be courteous to those still taking the exam.

Good luck.

Timing tip. A rate of 2.5 minutes per multiple choice problem leaves 5 minutes to go over any parts of the exam you might have skipped.



```
#include<iostream>
using std::cout; using std::endl;
#include<vector>
using std::vector;
#include<algorithm>
using std::sort; using std::accumulate;

long fn1(vector<vector<long>>& v, long x){
  long temp = x;
  for (auto i=v.begin(); i!= v.end(); i++)
    if ( (*i)[0] > temp )
      temp = (*i)[0];        // Line 1
  return temp;
}

// sort works on vectors as on strings: compare across vectors
// element by element.
long fn2(vector<vector<long>>& v){
  sort(v.begin(), v.end());

  // Line 2 below
  auto result = accumulate(v[0].begin(), v[0].end(), 0);
  return result;
}

int main (){
  vector<vector<long>> v = { {4,5,6}, {1,2,3}, {0,8,9} };
  cout << fn1(v,5) << endl;    // Line 3
  cout << v[0][0] << endl;     // Line 4
  cout << fn2(v) << endl;      // Line 5
  cout << v[0][0] << endl;     // Line 6
}
```

Figure 1

1) What type is `(*i)[0]` on `Line 1` of Figure 1?
   a) `vector<vector<long>>`
   b) `vector<long>`
   c) `vector<long>::iterator`
   d) `long*`
   e) None of the above.

2) What type is returned by `v[0].begin()` on `Line 2` of Figure 1?
   a) `vector<vector<long>>`
   b) `vector<long>`
   c) `vector<long>::iterator`
   d) `long*`
   e) None of the above.

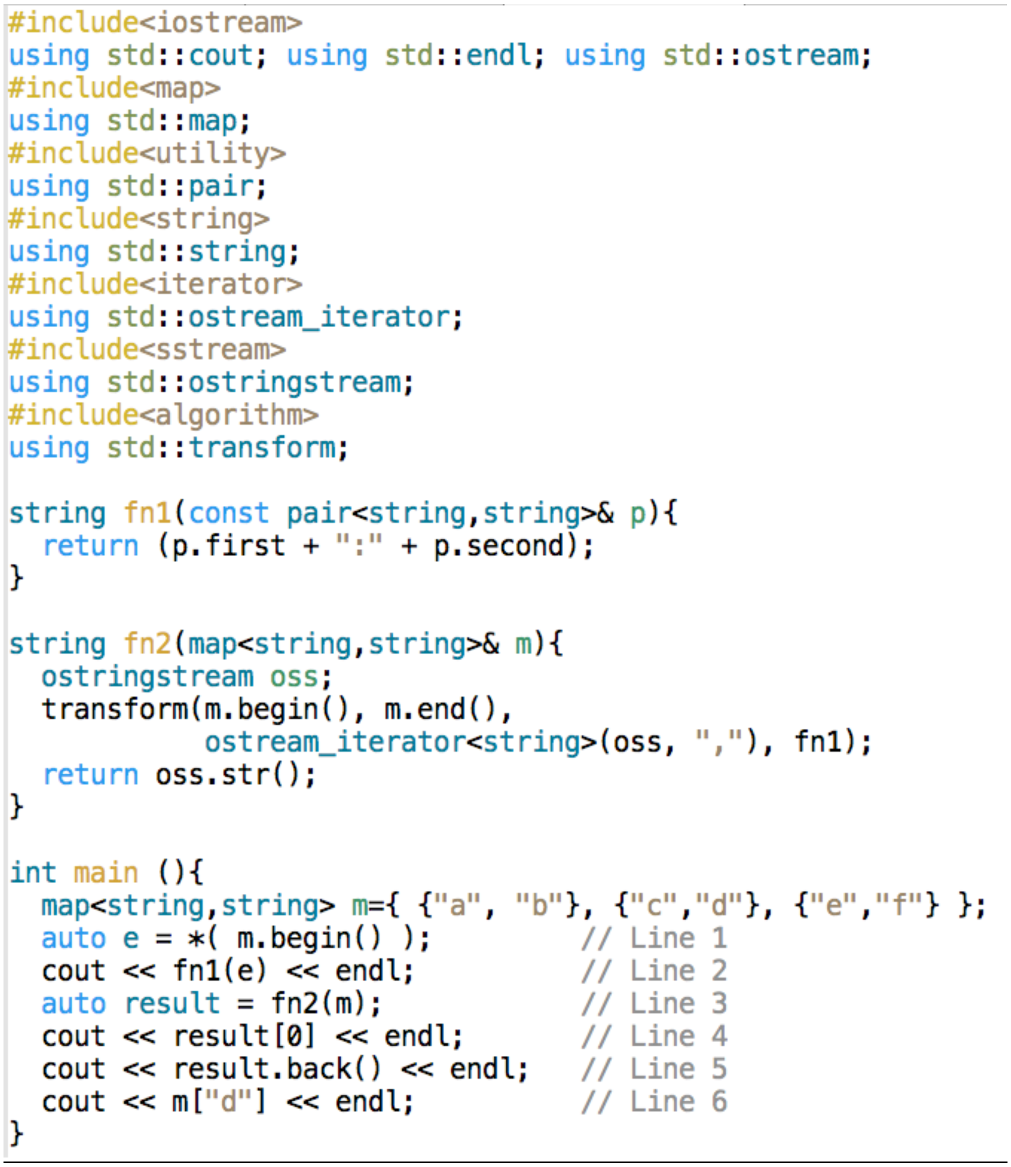
3) What output is produced by `Line 3` in Figure 1?
   a) `1`
   b) `2`
   c) `3`
   d) `4`
   e) None of the above.

4) What output is produced by `Line 4` in Figure 1?
   a) `1`
   b) `2`
   c) `3`
   d) `4`
   e) None of the above.

5) What output is produced by `Line 5` in Figure 1?
   a) `15`
   b) `6`
   c) `17`
   d) `38`
   e) None of the above.

6) What output is produced by `Line 6` in Figure 1?
   a) `1`
   b) `2`
   c) `3`
   d) `4`
   e) None of the above.

7) Which of the following are true about a C++ `lambda`?
   a) it has no name
   b) can act as a function
   c) is commonly used as part of a generic algorithm
   d) All of the above
   e) None of the above
8) Which of the following is ***not*** a method of `map`?
   a) `find`
   b) `size`
   c) `push_front`
   d) `count`
   e) None of the above
9) What is the proper interpretation of `var1->y ?`
   a) return the data member `y` of `var1`
   b) dereference the pointer `var1` and return its data member `y`
   c) dereference the pointer `y` and return its value
   d) set var1 to y
   e) None of the above.
10) What is meaning of the `cin.ignore(100)`?
   a) ignore all input from cin except for the value 100.
   b) prevent cin from getting the value 100 from the input stream
   c) empty the cin buffer of 100 inputs stored there
   d) stop input for the next 100 milliseconds
   e) None of the above
11) Which of the following are true about the variable `string::npos`
   a) It is the maximum number of positions in a string
   b) In a substring operation, it signifies "from the beginning of the string"
   c) It is returned when a find operation fails on a string
   d) All of the above
   e) None of the above
12) Which of the following constitutes a "to conversion" for a class named `MyClass`?
   a) `MyClass(string)`
   b) `MyClass(long, long)`
   c) `MyClass(MyClass&)`
   d) All of the above
   e) None of the above
13) Which of the following commands allocates dynamic memory during program runtime?
   a) `copy`
   b) `add`
   c) `create`
   d) `new`
   e) None of the above



```cpp
#include<iostream>
using std::cout; using std::endl; using std::ostream;
#include<map>
using std::map;
#include<utility>
using std::pair;
#include<string>
using std::string;
#include<iterator>
using std::ostream_iterator;
#include<sstream>
using std::ostringstream;
#include<algorithm>
using std::transform;

string fn1(const pair<string,string>& p){
  return (p.first + ":" + p.second);
}

string fn2(map<string,string>& m){
  ostringstream oss;
  transform(m.begin(), m.end(),
            ostream_iterator<string>(oss, ","), fn1);
  return oss.str();
}

int main (){
  map<string,string> m={ {"a", "b"}, {"c","d"}, {"e","f"} };
  auto e = *( m.begin() );         // Line 1
  cout << fn1(e) << endl;          // Line 2
  auto result = fn2(m);            // Line 3
  cout << result[0] << endl;       // Line 4
  cout << result.back() << endl;   // Line 5
  cout << m["d"] << endl;          // Line 6
}
```

Figure 2

14) For the program in Figure 2, what type is `e` in `Line 1`.

a) `string`
b) `map<string, string>`
c) `map<string,string>::iterator`
d) `pair<string,string>::iterator`
e) None of the above

15) For the program in Figure 2, give the output of `Line 2`

a) `b`
b) `a:b`
c) `ab`
d) `a`
e) None of the above

16) For the program in Figure 2, what type is `result` in `Line 3`.

a) `string`
b) `map<string, string>`
c) `map<string,string>::iterator`
d) `pair<string,string>::iterator`
e) None of the above

17) For the program in Figure 2, give the output of `Line 4`.

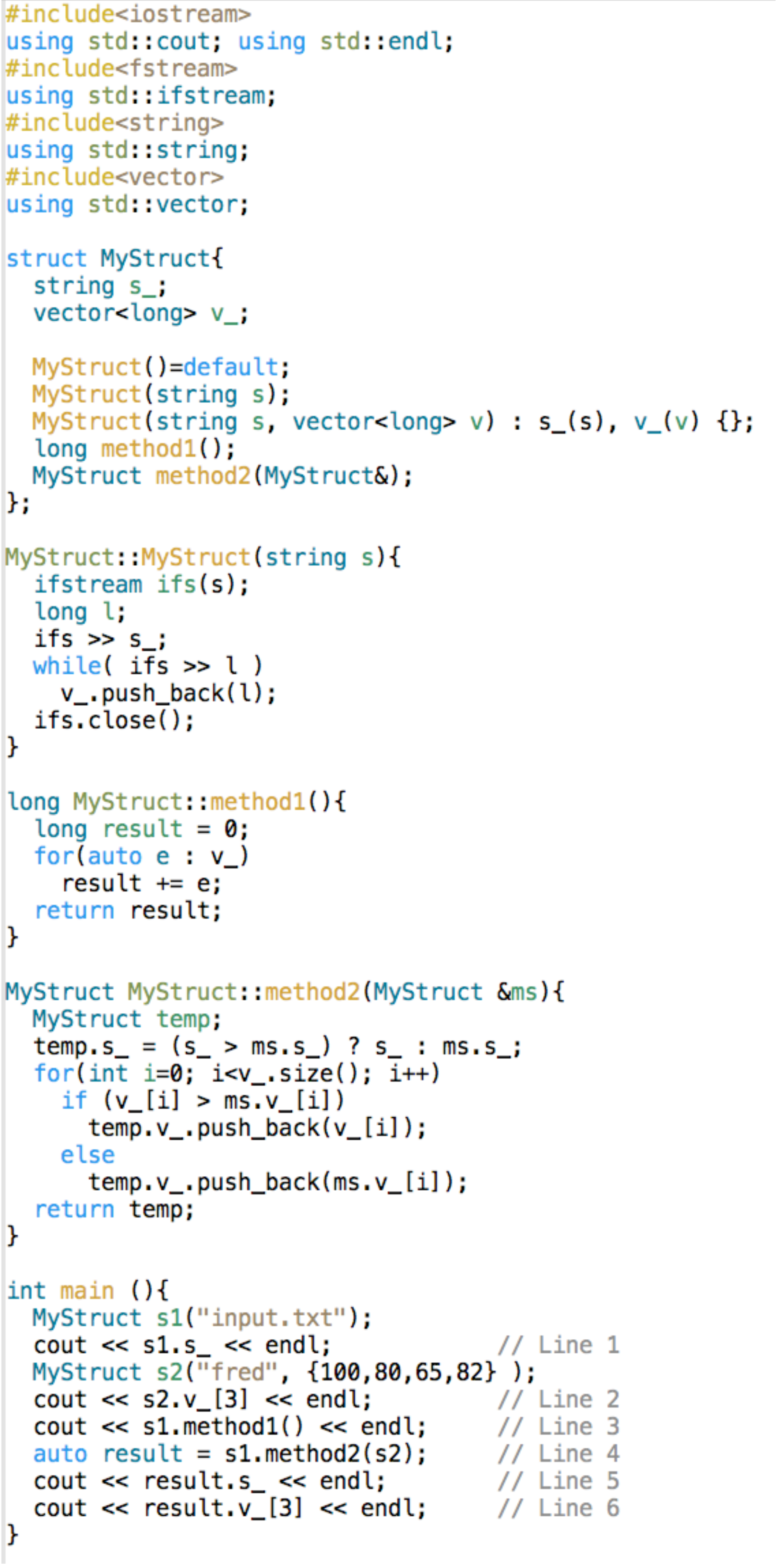
a) `b`
b) `a:b`
c) `ab`
d) `a`
e) None of the above

18) For the program in Figure 2, give the output of `Line 5`.

a) `e`
b) `f`
c) `e:f`
d) `ef`
e) None of the above

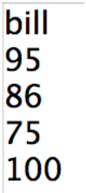
19) For the program in Figure 2, give the output of `Line 6`.

a) `d`
b) `c`
c) `empty string`
d) error, line will not compile!
e) None of the above



```cpp
#include<iostream>
using std::cout; using std::endl;
#include<fstream>
using std::ifstream;
#include<string>
using std::string;
#include<vector>
using std::vector;

struct MyStruct{
  string s_;
  vector<long> v_;

  MyStruct()=default;
  MyStruct(string s);
  MyStruct(string s, vector<long> v) : s_(s), v_(v) {};
  long method1();
  MyStruct method2(MyStruct&);
};

MyStruct::MyStruct(string s){
  ifstream ifs(s);
  long l;
  ifs >> s_;
  while( ifs >> l )
    v_.push_back(l);
  ifs.close();
}

long MyStruct::method1(){
  long result = 0;
  for(auto e : v_)
    result += e;
  return result;
}

MyStruct MyStruct::method2(MyStruct &ms){
  MyStruct temp;
  temp.s_ = (s_ > ms.s_) ? s_ : ms.s_;
  for(int i=0; i<v_.size(); i++)
    if (v_[i] > ms.v_[i])
      temp.v_.push_back(v_[i]);
    else
      temp.v_.push_back(ms.v_[i]);
  return temp;
}

int main (){
  MyStruct s1("input.txt");
  cout << s1.s_ << endl;              // Line 1
  MyStruct s2("fred", {100,80,65,82} );
  cout << s2.v_[3] << endl;           // Line 2
  cout << s1.method1() << endl;       // Line 3
  auto result = s1.method2(s2);       // Line 4
  cout << result.s_ << endl;          // Line 5
  cout << result.v_[3] << endl;       // Line 6
}
```

Contents of file `input.txt`



```
bill
95
86
75
100
```

Figure 3

20) For the program in Figure 3, give the output of `Line 1`?

a) `bill`
b) `fred`
c) empty string
d) `100`
e) None of the above

21) For the program in Figure 3, give the output of `Line 2`?

a) `65`
b) `82`
c) `75`
d) `100`
e) None of the above

22) For the program in Figure 3, give the output of `Line 3`?

a) `356`
b) `95`
c) `85`
d) `76`
e) None of the above

23) For the program in Figure 3, what type is `result` on `Line 4`?

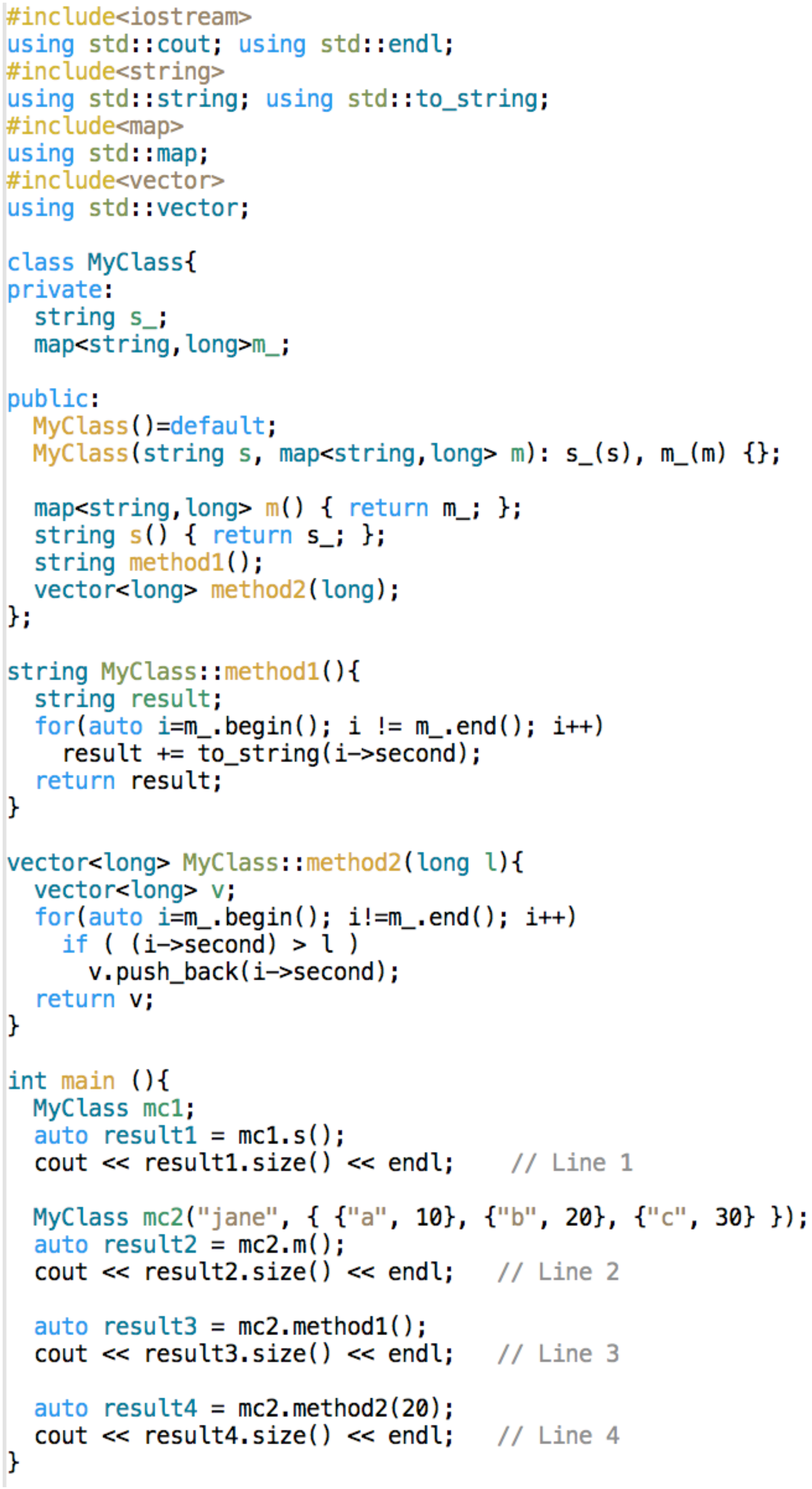
a) `MyStruct`
b) `MyStruct *`
c) `MyStruct &`
d) `long`
e) None of the above

24) For the program in Figure 3, give the output of `Line 5`?

a) `bill`
b) `fred`
c) empty string
d) `100`
e) None of the above

25) For the program in Figure 3, give the output of `Line 6`?

a) `65`
b) `82`
c) `75`
d) `100`
e) None of the above



```cpp
#include<iostream>
using std::cout; using std::endl;
#include<string>
using std::string; using std::to_string;
#include<map>
using std::map;
#include<vector>
using std::vector;

class MyClass{
private:
  string s_;
  map<string,long>m_;

public:
  MyClass()=default;
  MyClass(string s, map<string,long> m): s_(s), m_(m) {};

  map<string,long> m() { return m_; };
  string s() { return s_; };
  string method1();
  vector<long> method2(long);
};

string MyClass::method1(){
  string result;
  for(auto i=m_.begin(); i != m_.end(); i++)
    result += to_string(i->second);
  return result;
}

vector<long> MyClass::method2(long l){
  vector<long> v;
  for(auto i=m_.begin(); i!=m_.end(); i++)
    if ( (i->second) > l )
      v.push_back(i->second);
  return v;
}

int main (){
  MyClass mc1;
  auto result1 = mc1.s();
  cout << result1.size() << endl;    // Line 1

  MyClass mc2("jane", { {"a", 10}, {"b", 20}, {"c", 30} });
  auto result2 = mc2.m();
  cout << result2.size() << endl;   // Line 2

  auto result3 = mc2.method1();
  cout << result3.size() << endl;   // Line 3

  auto result4 = mc2.method2(20);
  cout << result4.size() << endl;   // Line 4
}
```

Figure 4

26) For the program in Figure 4, what value is printed by `Line 1`?

a) `1`
b) `2`
c) `3`
d) `4`
e) None of the above

27) For the program in Figure 4, what value is printed by `Line 2`?

a) `1`
b) `2`
c) `3`
d) `4`
e) None of the above

28) For the program in Figure 4, what value is printed by `Line 3`?

a) `1`
b) `2`
c) `3`
d) `4`
e) None of the above

29) For the program in Figure 4, what value is printed by `Line 4`?

a) `1`
b) `2`
c) `3`
d) `4`
e) None of the above

30) For the program in Figure 4, which of the following methods of `MyClass` could be properly called "accessors"?

a) `m`
b) `method1`
c) `method2`
d) `MyClass`
e) None of the above