

## Spring Semester, Dr. Punch. Exam #2 (3/29), 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;

size_t fn1(vector<long> &v1, vector<long> &v2){
    size_t i; // Line 1
    if (v1.size() > v2.size() ){
        for(i=v2.size(); i<v1.size(); ++i)
            v2.push_back(v1[i]);
    }
    else{
        for(i=v1.size(); i<v2.size(); ++i)
            v1.push_back(v2[i]);
    }
    return i;
}

void fn2 (vector<long> &v, long num){
    long var;
    if (num < v.size() )
        var = v[num];
    else
        var = 1;

    for(int i=0; i<var; ++i){
        v.push_back(v[0]);
        v.erase(v.begin() );
    }
}

int main () {
    vector<long> v1{1,2,3};
    vector<long> v2{7,8,9,10};

    cout << fn1(v1, v2) << endl; // Line 2
    cout << v1.front() << endl; // Line 3
    cout << v1.back() << endl; // Line 4

    vector<long> v3{11,12,13,14};
    fn2(v3,2);
    cout << v3.front() << endl; // Line 5

    vector<long> v4{4,5,6,7};
    fn2(v4, 6);
    cout << v4.front() << endl; // Line 6
}

```

Figure 1

- 1) Which of the following are true about the `size_t` type on Line 1 of Figure 1?
- a) It is an unsigned type
  - b) It can hold the maximum size of any container
  - c) The compiler generates a warning when comparing it against a `long` or `int`
  - d) All of the above
  - e) None of the above.
- 2) What output is produced by Line 2 of Figure 1?
- a) 1
  - b) 4
  - c) 7
  - d) 10
  - e) None of the above.
- 3) What output is produced by Line 3 in Figure 1?
- a) 1
  - b) 4
  - c) 7
  - d) 10
  - e) None of the above.
- 4) What output is produced by Line 4 in Figure 1?
- a) 1
  - b) 4
  - c) 7
  - d) 10
  - e) None of the above.
- 5) What output is produced by Line 5 in Figure 1?
- a) 11
  - b) 12
  - c) 13
  - d) 14
  - e) None of the above.
- 6) What output is produced by Line 6 in Figure 1?
- a) 4
  - b) 5
  - c) 6
  - d) 7
  - e) None of the above.

7) Which of the following are true about the `explicit` designation on a method?

- a) Used to prevent implicit conversion.
- b) Indicates permission to access private class elements
- c) Method spelling must be explicit.
- d) All of the above
- e) None of the above

8) What is the type returned by a `map<long, string> insert`?

- a) `long`
- b) `long&`
- c) `string`
- d) `bool`
- e) None of the above

9) Which of the following are true about a lambda function?

- a) it is a nameless function.
- b) they are often used in conjunction with STL algorithms
- c) they have a capture list
- d) All of the above
- e) None of the above

10) Which of the following are true about C++ exceptions?

- a) Potential elements that might throw are wrapped in a try block
- b) If an error is thrown, control moves to the catch block of the same error type.
- c) An uncaught exception will halt the program
- d) All of the above
- e) None of the above

11) Which of the following are true about the operator `*`?

- a) As a binary operation it represents multiply.
- b) In a declaration it represents a reference type.
- c) As a unary operation it represents `sizeof`.
- d) All of the above
- e) None of the above

12) Which of the following are true about the special variable `this`?

- a) The programmer can **directly** assign it a new value in a class's method.
- b) The programmer must set it to use it in a class method
- c) It is a pointer type.
- d) All of the above
- e) None of the above

```

#include<iostream>
using std::cout; using std::endl;
#include<vector>
using std::vector;
#include<algorithm>
using std::copy; using std::sort;
#include<numeric>
using std::accumulate;
#include<iterator>
using std::back_inserter;

long fn1(long t, long e){
    return e*e + t;
}

long fn2 (vector<long> v){
    return accumulate (v.begin(), v.end(), 0, fn1);
}

vector<long> fn3(vector<long> v, long val){
    vector<long> result;
    sort(v.begin(), v.end() );
    copy(v.begin(), v.begin()+3, back_inserter(result) );
    return result;
}

long fn4(vector<vector<long>> v){
    long result = 0;
    for (auto r : v){           // Line 1
        for (auto c : r){
            if (c % 2)
                result += c;
        }
    }
    return result;
}

int main (){
    vector<long> v1{3,4,1,2};
    cout << fn2(v1) << endl;           // Line 2
    auto val = fn3(v1,3);              // Line 3
    cout << val.size() << endl;        // Line 4
    cout << val.back() << endl;       // Line 5
    |
    vector<vector<long>> v2{ {1,2,3}, {4,5,6}, {7,8,9} };
    cout << fn4(v2) << endl;          // Line 6
}

```

Figure 2

13) For the program in Figure 2, what type is r on Line 1.

- a) long
- b) vector<long>
- c) vector<long>::iterator
- d) vector<vector<long>>
- e) None of the above

B

14) What output is produced by Line 2 in Figure 2?

- a) 10
- b) 24
- c) 30
- d) 4
- e) None of the above

C

15) For the program in Figure 2, what type is var in Line 3?

- a) long
- b) vector<long>
- c) vector<long>::iterator
- d) vector<vector<long>>
- e) None of the above

B

16) What output is produced by Line 4 in Figure 2?

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

C D

17) What output is produced by Line 5 in Figure 2?

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

C B

18) What output is produced by Line 6 in Figure 2?

- a) 6
- b) 15
- c) 24
- d) 45
- e) None of the above

E

```

#include<iostream>
using std::cout; using std::endl;
#include<map>
using std::map;
#include<vector>
using std::vector;
#include<string>
using std::string;
#include<sstream>
using std::ostringstream;

long fn1(map<string, vector<long>> &m){
    long result;
    for (auto &p : m){
        result = 0;
        for (auto l : p.second){
            result += l;
        }
        p.second.push_back(result);
    }
    return result;
}

bool fn2(map<string, vector<long>> &m, string s, long lng){
    auto i = m.find(s);
    if (i == m.end() )
        auto result = m.insert({s, {lng, lng, lng} });
    else
        m[s] = {lng,lng};
    return i == m.end();
}

string fn3(map<string, vector<long>> &m){
    ostringstream oss;
    for (auto p : m)
        for (auto l : p.second)
            oss << l;
    return oss.str();
}

int main (){
    map<string, vector<long>> m1{ {"a", {1,2,3}}, {"b", {3,2,1}} };
    cout << fn1(m1) << endl;           // Line 1
    cout << m1["a"].back() << endl;    // Line 2
    cout << m1["b"].size() << endl;    // Line 3

    map<string, vector<long>> m2{ {"c", {1,2,1}}, {"d", {3,2,3}} };
    cout << fn2(m2, "d", 5) << endl;   // Line 4
    cout << m2.size() << endl;         // Line 5
    cout << m2["d"][1] << endl;        // Line 6

    map<string, vector<long>> m3{ {"e", {5,6,7}}, {"f", {7,6,5}} };
    cout << fn3(m3) << endl;          // Line 7
}

```

Figure 3

19) What output is produced by Line 1 in Figure 3?

- a) 1
- E** b) 2
- c) 3
- d) 4
- e) None of the above

20) What output is produced by Line2 in Figure 3?

- a) 1
- E** b) 2
- c) 3
- d) 4
- e) None of the above

21) What output is produced by Line 3 of Figure 3?

- a) 6
- C** b) 5
- c) 4
- d) 3
- e) None of the above

22) What output is produced by Line 4 of Figure 3?

- a) 0
- A** b) 1
- c) true
- d) false
- e) None of the above

23) What output is produced by Line 5 of Figure 3?

- a) 1
- B** b) 2
- c) 3
- d) 4
- e) None of the above

24) What output is produced by Line 6 of Figure 3?

- a) 6
- B** b) 5
- c) 4
- d) 3
- e) None of the above

25) What output is produced by Line 7 of Figure 3?

- a) 18
- E** b) 36
- c) 567
- d) 765
- e) None of the above



<pre> #include&lt;iostream&gt; using std::cout; using std::endl; #include&lt;string&gt; using std::string; #include&lt;utility&gt; using std::pair; using std::make_pair;  struct MyStruct{     string str1="";     string str2="";      MyStruct()=default;     MyStruct(string s1, string s2) : str1(s1), str2(s2) {} ;     pair&lt;string, size_t&gt; m1();     string m2();     void m3(char, long); };  pair&lt;string, size_t&gt; MyStruct::m1(){     if (str1.size() &gt; str2.size() )         return make_pair(str1, str1.size());     else         return make_pair(str2, str2.size()); }  string MyStruct::m2(){     size_t sz;     string result="";     sz = (str1.size() &lt; str2.size() ) ? str1.size() : str2.size();     for (size_t i=0; i&lt;sz; ++i){         result += str1[i];         result += str2[i];     }     return result; } </pre>	<pre> void MyStruct::m3(char ch, long lng){     string s1;     if (ch == 'r'){         for(int i = 0; i&lt;lng; ++i){             s1 = str1.back();             str1 = str2.front() + str1.substr(0,str1.size() - 1 );             str2 = str2.substr(1) + s1;         }     } // of if     else{         for(int i = 0; i&lt;lng; ++i){             s1 = str1.front();             str1 = str1.substr(1)+ str2.back();             str2 = s1 + str2.substr(0, str2.size() - 1);         } // of else     } }  int main (){     MyStruct struct1("abcde", "123");     auto p = struct1.m1();     cout &lt;&lt; p.first &lt;&lt; endl;    // Line 1     cout &lt;&lt; p.second &lt;&lt; endl;  // Line 2      cout &lt;&lt; struct1.m2() &lt;&lt; endl; // Line 3      MyStruct struct2("lmnop", "67890");     struct2.m3('l', 1);     cout &lt;&lt; struct2.str1 &lt;&lt; endl; // Line 4     cout &lt;&lt; struct2.str2 &lt;&lt; endl; // Line 5 } </pre>
--	--

Figure 4

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

- a) abcde
- b) 123
- c) 3
- d) 5
- e) None of the above

A

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

E

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

- a) abcde
- b) 123
- c) a1b2c3de
- d) a1b2c3
- e) None of the above

D

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

- a) lmnop
- b) mnopl
- c) 0mnop
- d) mnop0
- e) None of the above

D

30) For the program in Figure 4, what value is printed by Line 5?

- a) 67890
- b) 06789
- c) 16789
- d) 6789p
- e) None of the above

C