1. What will be printed when executing the following code?

#include <iostream>  
  
int main(){  
 std::out<< 25u – 50;  
 return 0;  
}

A. 25  
B. -25  
C. -75  
D. other value  
E. Nothing, there is an error

Correct Solution: D.  
Explanation: 50 is turned to 50u, then the result -25 is unsigned as well so we get 4294967271.

2. What is wrong with the following code?

my\_struct \*bar;  
/\* some code \*/  
memset(bar, 0, sizeof(bar)); // this set all memory of bar to 0.

A. The last line will not do what we want it to do  
B. bar might be set to NULL  
C. Nothing is wrong  
D. Answers A and B are correct

Correct Solution: A.  
Explanaion: to fix the code we need to write “sizeof(\*bar)” and if bar is NULL it doesn’t matter since “sizeof” is a compile time construct.

3. What are the values of ‘a’ and ‘b’ at the end?

int a = 5;  
int b = a++;  
b = ++a;  
a = b--;  
  
A. a=5, b=5  
B. a=5, b=6  
C. a=6, b=5  
D. a=6, b=6  
E. a=6, b=7  
F. a=7, b=6  
G. a=7, b=7

Correct Solution: F.

4. The following code… ?

size\_t s = buf->size();  
while( --sz >= 0 ){  
 /\* some code \*/  
}

A. might crash in the first line  
B. might never enter the while loop  
C. might never end the while loop  
D. the code has no problems  
E. answers A and B are correct  
F. answers A and C are correct

Correct Solution: F.  
Explanation: size\_t is unsigned thus the while loop will never end. if buf is NULL the first line will crash.

5. What will be printed?

#include <iostream>  
  
Class A {  
public:  
 A() {}  
 ~A(){ throw 42; }  
};  
  
int main(){  
 try{  
 A a;  
 throw 32;  
 } catch( int a){  
 std::cout << a;  
 }  
}

A. 32  
B. 42  
C. nothing  
D. the program will crash  
E. it can’t be determined

Correct solution: D.  
Explanation: when handling “throw 32”, another exception is thrown when calling ~A().

6. What is the output of the following code?

#include <iostream>  
  
int main(){  
 int a[] = {1,2,3,4,5,6};  
 std::cout << (1+3)[a] – a[0] + (a+1)[2];  
}

A. 8  
B. 6  
C. 9  
D. 5

Correct Solution: A.  
Explanation: arithmetics.

7. How many times will the loop execute?

unsigned char half\_time = 150;  
for( unsigned char i = 0; i < 2\*half\_time; ++i){  
 /\* some code \*/  
}

A. 300  
B. 600  
C. 150  
D. 0  
E. forever  
D. the program will crash before reaching the loop

Correct Solution: E.  
Explanation: ‘2\*half\_time’ is int with a value of 300. ‘i’ is unsigned char with a max value of 255 thus never reaching the end condition.

8. What will be printed?

#include <iostream>

struct A

{

int data[2];

A(int x, int y) : data{x, y} {}

virtual void f() {}

};

int main(int argc, char \*\*argv)

{

A a(22, 33);

int \*arr = (int \*) &a;

std::cout << arr[2] << std::endl;

return 0;

}  
   
A. 22  
B. 33  
C. Nothing  
D. Random string  
E. It can’t be determined

Correct Solution: E.

Explanation: if the architecture is 32bit 33 is printed, and if it’s 64bit then 22 is printed. This is because of the size of the vptr instered.