1. /* -----
 * Definition & Declaration
 * 编译运行以下 C++代码，哪一个会报错？*/

```
/* A */
int bar(int x);
int bar(double x);

/* B */
int foo();
int foo(){
  return 0;
}

/* C */
extern string b;
string b = "PPP";

/* D */
int a;
int a = 1;
```

2. /* -----
 * Scope
 * 程序运行的结果为？*/

```
int x = 0;
void foo(){
  int x = 1;
  std::cout << x << std::endl;
}

int main(){
  int x = 2;
```

```
  foo();
  std::cout << x << std::endl;
  for (int x = 3; x < 4; ++x){
    std::cout << x << std::endl;
  }
}
```

A. 1 0 3

B. 2 0 2

C. 1 2 3

D. 0 2 2

3.

```
/* -----
 * Call by value/reference/const-reference
 * 程序运行的结果为？*/
int foo(int x){
  x += 1;
  return x;
}

int bar(int &y){
  y += 1;
  return y;
}

int main(){
  int x = 0;
  int a = foo(x);
  std::cout << x << std::endl;
  std::cout << a << std::endl;

  int y = 10;
  int b = bar(y);
  std::cout << y << std::endl;
```

```
  std::cout << b << std::endl;

}
```

A. 0 1 11 11

B. 1 1 11 11

C. 1 1 10 11

D. 0 1 10 11

4.(1)/* -----

 * Reference

 * 程序运行的结果为？*/

```
int main(){
  int i = 1;
  int &r = i;
  std::cout << r << std::endl;

  r++;
  std::cout << i << std::endl;

  const int &cr = i;
  i = 5;
  std::cout << r << std::endl;
  std::cout << cr << std::endl;
}
```

A. 1 1 5 5

B. 1 2 2 5

C. 1 1 2 5

D. 1 2 5 5

(2)若在结尾处添加：

```
cr = 10;
std::cout << i << std::endl;
```

程序的运行结果为？

A. 再输出 10

B. 再输出 5

C. 报错

5、请问以下代码的输出结果是什么？

```
class point {
public:
    point(int a, int b) {x = a; y = b;}
    point(const point &p) {x = 2 * p.x; y = 2 * p.y;}
    void print() {cout << x << " " << y << endl;}
private:
    int x;
    int y;
};

int main() {
    point p1(10, 20);
    point p2(p1);
    point p3 = p1;
    point p4(1, 2);
    p1.print();
    p2.print();
    p3.print();
    p4.print();
    p4 = p1;
    p4.print();
}
```

A. 10 20

10 20

20 40

1 2

10 20

B. 10 20
20 40
10 20
1 2
10 20

C. 10 20
20 40
20 40
1 2
10 20

D. 10 20
20 40
20 40
1 2
20 40

6、以下声明有错的是？

A.

```
struct B {
    int **ptr;
    int *pi;
    int a;
};
```

B.

```
class A {
    int a;
    int b;
}
```

C.

```
class point {
    int x;
    int y;
    friend ostream & operator << (ostream &output, point & p);
};
```

D.

```
struct D {
    int x;
    void print();
};
```

7、请问哪种组合里的函数都可以使 main()输出正确的结果 5？

（注：比如答案(1)(2)(3)的意思是指用(1)或(2)或(3)的方式都能得到正确结果）

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

class HaHa {
private:
    int x;
public:

    HaHa(int a) {x = a;}
    int get_data() {return x;}
    (1)friend ostream & operator << (ostream &output, HaHa &haha) {output << haha.get_data(); return output;};
    (2)ostream & operator << (ostream &output) {output << this->x; return output;}
    (3)ostream & operator << (ostream &output, const HaHa &haha) {output << haha.get_data(); return output;}
};
```

(4)ostream & operator << (ostream &output, HaHa &haha) {

    output << haha.get_data();

    return output;

}

int main() {

    HaHa a(5);

    cout << a << endl;

}

A.(1)(2)(3)

B.(2)(3)(4)

C.(1)(4)

D.(1)(2)(4)

8、以下类的定义中，需要我们自己显式地编写析构函数的是：

A.

class A {

private:

    int a;

    int b;

public:

    A(int x, int y) {a = x; b = y;}

}

B.

class B {

private:

    int *ptr;

    int a;

public:

    B(int x) {a = x; ptr = new int(4);}

};

C.

```
class C {
private:
    int x;
    int y;
    friend ostream & operator << (ostream &output, point & p);
public:
    C(int &a, int &b) {x = a; y = b;}
};
```

D.

```
class D {
private:
    int x;
public:
    D(int *pi) {x = *pi;}
    void print();
};
```