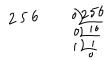
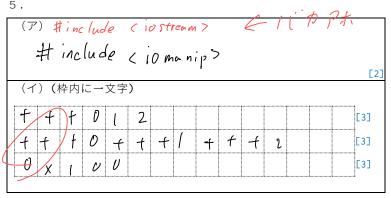
```
msg = "hello";
                                                           const int argc, const char[] agv [2]
                                                  [2]
   msg = msg + "hello";
                                                           inti-0; i < algc; itt
                                                            String str = arguli];
   leg th = msg. len();
                                                  [2]
                                                                                                           [2]
                                                               Str. reverse();
                                                                                                           [2]
                                                              reverse (str. begin(), str. end());
      (略)
     class Point {
       int x, y, z;
     public:
      friend istream & operator >> (istream & in, Point & p);
      friend ostveam& operator << (ostream& out, const Point & p);
[6]
     istream & operator >> (istream & in, Point & p) {
          in >> p. x >> p. y >> p.2;
          return inj
     ostream& operator << (ostream & out, const Point & p) {
[8]
          Out << p. x << ' ' << p. x << ' ' << p. 2 =
          return out;
      3
     int main() { (略) }
                                                          while (!in.eof()){
     #include <iostream>
     # include <fstream >
                                                              in> n >w;
(A)
[2]
                                                             total += static_cast cflat +> (n * w);
     using namespace std;
                                                    (C)
     int main()
                      // 各行の「個数」
       int n;
                      // 各行の「重さ」
       float w;
       float total = 0; // 総重量
                                                          Ofstream out;
       if stream in;
                                                     (Đ)
                                                          Out. open (" out put . tx+");
                                                     [2]
(B)
       in . open ("data.tx+");
                                                    (E)
                                                           out << total;
                                                     [2]
                                          (右上へ)
```





```
(ア) [BallFactory.cpp]
                                                        (イ) [BallApp.cpp]
#include "Ball.h"
#include "Ball Factory.h"
                                                       void BallApp::init()
                                                         balls.resize(100);
# include "Basker Ball. h"
                                                         for (auto& b : balls) {
Hinclude "Tennis Ball. h"
                                                           int r = Rand::range(0, 9); // 0〜9の乱数
#include "Base Ball.h"
                                                          if (r <1)
Ball * Ball Factory: Create (Pall: TYPE type) {
                                                               b = Ball Factory : Create (Ball: TYPE::BASKETBALL);
    switch (type) {
                                                          else if (r <4)
       Case BASKETBALL:
                                                              b = Ball Factory :: Create (Ball: TYPE::BASEBALL);
           teturn new Basket Ball;
                                                          e Ise
       case BASEBALL:
                                                             b = Ball Factory : Create (Ball: TYPE: TENNISBALL)
           return new Base Ball;
      Case TENNISBALL:
          re turn new Tennis Ball,
      default:
         re turn nullptr;
    refurn nullptri
                                                          b->init();
                                                  [10] }
```

8. 7 A I [4]

(ウ) かストリーム [2] (ナ) インハンハイト文字 [2] (ウ) map 連想 西であり		
	(イ) メンク・ル バイトナラ (ウ) まあ 料	あ2 万·1
	[2] Map 22	例しり [2]
(エ) 区切り文字 [2] (オ) ならってらん [2] (カ) ま命入ら家算子	(才) (力) (力) (方) (方) (方) (方) (方) (方) (方) (方) (方) (方	
区切り文字 [2] からってらん [2] (1) 本人が異算子	[2] 「To Con [2] 「中八 5 男子	[2]

#曲出

11.

```
#include <iostream>
Hindude Littl except?
using namespace std;
class difference_size error: public logic_error {
  difference_size_error(const string& e): logic_error(e) {}
class Vector {
  Vector operator+(const Vector& v) {
     if (size != v. size) {
       throw difference_size_error ("200x1)king #12 Nit in 1+1");
    Vector tmp(size);
    for (size_t i = 0; i < size; i++)</pre>
     tmp.vec[i] = vec[i] + v.vec[i];
  // 略
};
int main()
  Vector v1(10);
  Vector v2(8);
  Vector v3;
  v3 = v1 + v2;
  Catch (const exception Re) {
     cout << c. what() << end);
  // 以降何かの処理
```

```
空欄(A)
[3] gu: speak();
空欄(B)
[3] Using Namespace tyoki;
空欄(C)
[3] pa: speak();
```