```cpp
/*
//-----------B站(bilibili.com)课程“从C到C++快速入门(2019版)”-------
//--------------https://www.bilibili.com/video/av40959422-----------
//-------------------源代码文件---------------
//--------------请关注：------------
//youtube：hwdong
//博客：https://hwdong.net或https://hwdong-net.github.io
//腾讯课堂：http://hwdong.ke.qq.com
//B站：hw-dong

*/
#if 0
#define _CRT_SECURE_NO_WARNINGS
#include <cstdio>  //标准输入输出函数
#include <cmath>
#include <cstring> //字符串处理函数

int main() {
    printf("hello\n");
    double x = 3.14;
    printf("%lf %lf\n", sqrt(x), sin(x));

    char s[10] = "hello";
    puts(s);
    char s2[16];
    strcpy(s2, "world");
    puts(s2);
    strcat(s2, "sdfsdf");
    puts(s2);
    printf("%d %d\n", strlen(s), strlen(s2));
    return 0;
}
#endif

#if 0
#define _CRT_SECURE_NO_WARNINGS
#include <cstdio>  //标准输入输出函数
#include <cstring> //字符串处理函数
#include <malloc.h>
int main() {
# if 1
    char s[10];
    strcpy(s, "hello");
    puts(s);
#else
    char *s = (char *)malloc(12 * sizeof(char));
    strcpy(s, "hello world");
    puts(s);
#endif

}
#endif

```

```
54  #if 0
55
56  #include <iostream>  //C++标准输入输出流头文件
57  using namespace std;
58  int main() {
59      cout << "hello world!"<< endl;
60      cout << "https://a.hwdong.com" << endl;
61      cout << 3+4 << endl;
62  #if 0
63      double radius;
64      std::cin >> radius; //标准输入流对象cin 输入运算符>>
65      cout << 3.14*radius*radius;
66  #endif
67      std::cout << "  *\n";
68      std::cout << " *  *\n";
69      std::cout << " *  *\n";
70
71
72      return 0;
73  }
74  #endif
75
76  #if 0
77  #include <iostream>
78  using namespace std;
79  void help() {
80      cout << "=========简单计算器=========\n";
81      cout << "请输入：左运算数  运算符  右运算符\n";
82  }
83
84  int main() {
85      while (1) {
86          help();
87          double a, b;
88          char op;
89          cin >> a >> op >> b;
90          if (op == '+')
91              cout << a + b << endl;
92          //...补充你的代码
93      }
94  }
95  #endif
96
97  #if 0
98  #include <fstream>
99  #include <iostream>
100 #include <string>
101 using namespace std;
102 int main() {
103     ofstream oF("test.txt");
104     oF << 3.14 << " " << "hello world\n";
105     oF.close();
106     ifstream iF("test.txt");
```

```
107     double d;
108     string str;
109     iF >> d >> str;
110     cout<<d <<" "<< str<<endl;
111 
112     return 0;
113 }
114 #endif
115 
116 #if 0
117 #include <iostream>
118 using namespace std;
119 
120 int main() {
121     int a = 3, &r = a;
122     cout << a << '\t' << r << endl;
123     r = 5;
124     cout << a << '\t' << r << endl;
125     return 0;
126 }
127 
128 #endif
129 
130 #if 0
131 #include <iostream>
132 using namespace std;
133 void swap(int x, int y) {
134     cout << x << '\t' << y << endl;
135     int t = x;
136     x = y;
137     y = t;
138     cout << x << '\t' << y << endl;
139 }
140 
141 int main() {
142     int a = 3, b = 4;
143     cout << a << '\t' << b << endl;
144     swap(a, b);
145     cout << a << '\t' << b  << endl;
146 }
147 #endif
148 
149 #if 0
150 void swap(int *x, int *y) {
151     int t = *x;
152     *x = *y;
153     *y = t;
154 }
155 #include <iostream>
156 using namespace std;
157 int main() {
158     int a = 3, b = 4;
159     cout << a << '\t' << b << endl;
```

```
160     swap(&a, &b);
161     cout << a << '\t' << b << endl;
162 }
163 #endif
164
165 #if 0
166 void swap(int &x, int &y) {
167     int t = x;
168     x = y;
169     y = t;
170 }
171 #include <iostream>
172 using namespace std;
173 int main() {
174     int a = 3, b = 4;
175     cout << a << '\t' << b << endl;
176     swap(a, b);
177     cout << a << '\t' << b << endl;
178 }
179 #endif
180
181
182 #if 0
183 #include <iostream>
184 using namespace std;
185 void print(char ch, int n = 1) {
186     for (int i = 0; i < n; i++)
187         cout << ch;
188 }
189 int main() {
190     print('*'); cout << endl;
191     print('*',3); cout << endl;
192     print('*',5); cout << endl;
193 }
194 #endif
195
196 #if 0
197 #include <iostream>
198 using namespace std;
199 int add(int x,int y=2,int z=3) {
200     return x + y + z;
201 }
202 int main() {
203     cout << add(5)<<endl;
204     cout << add(5,7) << endl;
205     cout << add(5,7,9) << endl;
206 }
207 #endif
208
209 #if 0
210 #include <iostream>
211 using namespace std;
212 int add(int x, int y = 2) {
```

```
213     return x + y ;
214 }
215 double add(double x, double y = 2.0) {
216     return x + y;
217 }
218 int main() {
219     cout << add(5,3) << endl;
220     cout << add(5.3, 7.8) << endl;
221     cout << add((double)5, 7.8) << endl;//歧义性
222 }
223 #endif
224
225 #if 0
226 #include <iostream>
227 using namespace std;
228 int add(int x, int y) {
229     return x + y;
230 }
231 double add(double x, double y ) {
232     return x + y;
233 }
234 int main() {
235     cout << add(5, 3) << endl;
236     cout << add(5.3, 7.8) << endl;
237 //  cout << add("hello", "world") << endl;
238 }
239 #endif
240
241 #if 0
242 #include <iostream>
243 #include <string>
244 using namespace std;
245
246 template<typename T>
247 T add(T x, T y) {
248     return x + y;
249 }
250 int main() {
251 #if 0
252     cout << add<int>(5, 3) << endl;
253     cout << add<double>(5.3, 7.8) << endl;
254     cout << add<int>(4, 6) << endl;
255     cout << add<string>("hello", "world") << endl;
256 #else
257     cout << add(5, 3) << endl;
258     cout << add(5.3, 7.8) << endl;
259     cout << add((double)5, 7.8) << endl; //歧义性
260 #endif
261 }
262 #endif
263
264
265 #if 0
```

```cpp
#include <iostream>
#include <string>
using namespace std;
int main() {
    string s = "hello", s2("world");
    //访问运算符.
    cout << s.size() << endl;
    string s3 = s.substr(1, 3);
    cout << s3<< endl;

    string s4 = s + " " + s2;
    cout << s4 << endl; //"hello world"

    s4[0] = 'H';
    s4[6] = 'X';
    cout << s4 << endl;

    int pos = s4.find("orl");
    cout << pos << endl;
    s4.insert(3, "ABCDE");
    cout << s4 << endl;

    for (int i = 0; i < s4.size(); i++)
        cout << s4[i] << "-";
    cout << "\n";

}
#endif

#if 0
#include <iostream>
using std::cout;
int main() {
    int arr[] = { 10,20,30,40 }; //大小固定，以后不能添加更多int值
    for (int i = 0; i < 4; i++)
        cout << arr[i] << '\t';
    cout << '\n';
}
#endif

#if 0
#include <iostream>
#include <vector>
using namespace std;
int main() {
    vector<int> v = { 7, 5, 16, 8 };
    //push_back()，最后添加一个元素
    v.push_back(25);
    v.push_back(13);

    //成员函数size()、下标运算符[]
    for (int i = 0; i < v.size(); i++)
        cout << v[i] << '\t';
```

```
319     cout << '\n';
320
321     v.pop_back();
322     for (int i = 0; i < v.size(); i++)
323         cout << v[i] << '\t';
324     cout << '\n';
325
326     v.resize(2);
327
328     for (int i = 0; i < v.size(); i++)
329         cout << v[i] << '\t';
330     cout << '\n';
331 }
332 #endif
333
334 #if 0
335 /*
336 指针就是地址，变量的指针就是变量的地址。
337 指针变量就是存储指针（地址）的变量。
338 */
339 #include <iostream>
340 using namespace std;
341 int main() {
342     int a=3;
343     int *p = &a;    //取地址运算符&用于获得a的地址：&a
344     cout << p << '\t' << &a << endl;
345     //取内容运算符*用于获得指针指向的变量(内存块)
346     cout << *p << '\t' << a << endl;      //*p就是a
347     *p = 5;                               //即a = 5;
348     cout << *p << '\t' << a << endl;
349 #if 1
350     int *q = p;              //q和p值相同，都是a的地址(指针)
351     cout << *p << '\t' << *q << '\t' << a << endl;
352     char *s = &a; //int *
353 #endif
354 }
355 #endif
356
357 #if 0
358 /*
359 用指针访问数组元素
360 */
361 #include <iostream>
362 using namespace std;
363 int main() {
364     int arr[] = { 10,20,30,40 };
365     int *p = arr;//数组名就是数组第一个元素的地址，即arr等于&(arr[0])
366     // p[i]就是*(p+i)
367     cout << *(p + 2) << '\t' << p[2] << '\t' << arr[2] << endl;
368
369     for (int *q = p + 4; p < q; p++)
370         cout << *p << '\t';
371     cout << '\n';
```

```
372 }
373 #endif
374
375 #if 0
376 /*
377 malloc free realloc
378   动态内存分配：new用于申请内存块、delete用于释放内存块
379   T *p = new T;
380   delete p;
381   T *q  = new T[5];
382   delete[] q;
383 */
384 #if 0
385 // 堆存储区
386 #include <iostream>
387 using namespace std;
388 int main() {
389     int *p = new int; //malloc
390     *p = 3;
391     cout << p << '\t' << *p << endl;
392     delete p;   //内存泄漏
393     p = new int;
394     *p = 5;
395     cout << p << '\t' << *p << endl;
396     delete p;
397 }
398 #else
399 #include <iostream>
400 using namespace std;
401 int main() {
402     int n = 4;
403     int *p = new int[n];
404     for (int i = 0; i < n; i++)
405         p[i] = 2 * i + 1;
406
407     for (int *q = p + n; p < q; p++)
408         cout << *p << '\t';
409     cout << '\n';
410
411     char *s = (char *)p;
412     char ch = 'A';
413     int n2 = n * sizeof(int) / sizeof(char);
414     for (int i = 0; i < n2; i++)
415         s[i] = ch + i;
416
417     for (char *r = s+n2; s < r; s++)
418         cout << *s;
419     cout << '\n';
420
421     delete[] p;
422 }
423 #endif
424 #endif
```

```
425
426 #if 0
427 /*
428 输入一组学生成绩(姓名和分数)，输出：平均成绩、最高分和最低分。
429 当然，也要能输出所有学生信息
430
431 */
432 #include <iostream>
433 #include <string>
434 #include <vector>
435 using namespace std;
436 struct student{
437     string name;
438     double score;
439     void print();
440 };
441 void student::print() {
442     cout << name << " " << score << endl;
443 }
444
445 int main() {
446 #if 0
447     student stu;
448     stu.name = "Li Ping";
449     stu.score = 78.5;
450     stu.print();
451 #endif
452     vector<student> students;
453
454     while (1) {
455         student stu;
456         cout << "请输入姓名 分数:\n";
457         cin >> stu.name >> stu.score;
458         if (stu.score < 0) break;
459         students.push_back(stu);
460     }
461     for (int i = 0; i < students.size(); i++)
462         students[i].print();
463
464     double min = 100, max=0, average = 0;
465     for (int i = 0; i < students.size(); i++) {
466         if (students[i].score < min) min = students[i].score;
467         if (students[i].score > max) max = students[i].score;
468         average += students[i].score;
469     }
470     average /= students.size();
471     cout << "平均分、最高分、最低分："
472         << average << " " << max << " " << min << endl;
473
474 }
475 #endif
476
477 #if 0
```

```
478 /*
479     this指针: 成员函数实际上隐含一个this指针。
480 */
481 #include <iostream>
482 #include <string>
483 using namespace std;
484
485 struct student {
486     string name;
487     double score;
488     void print() {
489         cout << this->name << " " << this->score << endl;
490     }
491 };
492 int main() {
493     student stu;
494     stu.name = "Li Ping";
495     stu.score = 78.5;
496     stu.print();   // print(&stu);
497 }
498 #endif
499
500
501
502 #if 0
503 /*
504     struct和class区别：
505     struct里的成员默认是public(公开的)
506     class里的成员默认是private(私有的)
507 */
508 #include <iostream>
509 #include <string>
510 using namespace std;
511
512 class student{
513 public: //接口
514     void print() {
515         cout << this->name << " " << this->score << endl;
516     }
517     string get_name() { return name; }
518     double get_score() { return score; }
519     void set_name(string n) { name = n; }
520     void set_score(double s) { score = s; }
521 private:
522     string name;
523     double score;
524 };
525 int main() {
526     student stu;
527
528 //  stu.name = "Li Ping";
529 //  stu.score = 78.5;
530     stu.set_name("Li Ping");
```

```
531     stu.set_score(78.5);
532     stu.print();    // print(&stu);
533     cout << stu.get_name() << " " << stu.get_score() << endl;
534 }
535 #endif
536
537 #if 0
538 /*
539 构造函数： 函数名和类名相同且无返回类型的成员函数。
540 */
541
542 #include <iostream>
543 #include <string>
544 using namespace std;
545
546 class student{
547     string name;
548     double score;
549 public:
550     student(string n,double s){ //不是默认构造函数
551         name = n; score = s;
552         cout << "构造函数\n";
553     }
554     void print() {
555         cout << this->name << " " << this->score << endl;
556     }
557 };
558 int main() {
559     student stu("LiPing",80.5); //在创建一个类对象时会自动调用称为“构造函数”的成员函数
560     stu.print();
561     student students[3];
562
563 }
564
565 #endif
566
567 #if 0
568 /* 运算符重载：针对用户定义类型重新定义运算符函数
569 */
570 #include <iostream>
571 #include <string>
572 using namespace std;
573 class student {
574     string name;
575     double score;
576 public:
577     student(string n, double s) {
578         name = n; score = s;
579     }
580     //友元函数
581     friend ostream& operator<<(ostream &o, student s);
582     friend istream& operator>>(istream &in, student &s);
```

```
583 };
584
585 ostream& operator<<(ostream &o, student s) {
586     cout << s.name << "," << s.score << endl;
587     return o;
588 }
589 istream& operator>>(istream &in, student &s) {
590     in >> s.name >> s.score;
591     return in;
592 }
593
594 int main() {
595     student stu("LiPing", 80.5);
596     cin >> stu; //operator>>(cin,stu)
597     cout << stu; //operator<<(cout,stu)
598 }
599
600 #endif
601
602 #if 0
603 #include <iostream>
604 #include <string>
605 using namespace std;
606
607 class Point{
608     double x, y;
609 public:
610     double operator[](int i) const{ //const函数
611         if (i == 0) return x;
612         else if (i == 1) return y;
613         else throw "下标非法!"; //抛出异常
614     }
615     double& operator[](int i) {
616         if (i == 0) return x;
617         else if (i == 1) return y;
618         else throw "下标非法!"; //抛出异常
619     }
620     Point(double x_,double y_) {
621         x = x_; y = y_;
622     }
623     Point operator+(const Point q) {
624         return Point(this->x+q[0],this->y + q[1]);
625     }
626
627     //友元函数
628     friend ostream & operator<<(ostream &o, Point p);
629     friend istream & operator>>(istream &i, Point &p);
630 };
631
632 ostream & operator<<(ostream &o, Point p) {
633     o <<p.x << " " << p.y<< endl;
634     return o;
635 }
```

```
636 istream & operator>>(istream &i, Point &p) {
637     i >> p.x >> p.y;
638     return i;
639 }
640 #if 0
641 Point operator+(const Point p,const Point q) {
642     return Point(p[0] + q[0], p[1] + q[1]);
643 }
644 #endif
645
646 int main() {
647     Point p(3.5, 4.8),q(2.0,3.0);
648 #if 0
649 //  cin >> p;
650     cout << p;
651     cout << p[0] << "-" << p[1] << endl; //p.operator[](0)
652     p[0] = 3.45; p[1] = 5.67;
653     cout << p;
654 #endif
655     cout << p<<q;
656     Point s = p + q; //p.operator+(q)  vs  operator+(p,q)
657     cout << s;
658 }
659 #endif
660
661 #if 0
662 #include <iostream>
663
664 using namespace std;
665
666 class String {
667     char *data; //C风格的字符串
668     int n;
669 public:
670     ~String() {
671         cout <<n<< " 析构函数!\n";
672         if(data)
673             delete[] data;
674     }
675 #if 1
676     String(const String &s) { //硬拷贝
677         cout << "拷贝构造函数!\n";
678         data = new char[s.n + 1];
679         n = s.n;
680         for (int i = 0; i < n; i++)
681             data[i] = s.data[i];
682         data[n] = '\0';
683     }
684 #endif
685     String(const char *s=0) {
686         cout << "构造函数!\n";
687         if (s == 0) {
688             cout << "s==0\n";
```

```
689             data = 0; n = 0; return;
690         }
691         const char *p = s;
692         while (*p) p++;
693         n = p - s;
694         data = new char[n + 1];
695         for (int i = 0; i < n; i++)
696             data[i] = s[i];
697         data[n] = '\0';
698     }
699     int size() { return n; }
700     char operator[](int i)const {
701         if (i<0 || i>=n ) throw "下标非法";
702         return data[i];
703     }
704     char& operator[](int i) {
705         if (i < 0 || i >= n) throw "下标非法";
706         return data[i];
707     }
708 };
709
710 ostream & operator<<(ostream &o, String s) {
711     for (int i = 0; i < s.size(); i++)
712         cout << s[i];
713     return o;
714 }
715 void f() {
716     String str,str2("hello world");
717     str2[1] = 'E';
718 //  cout << str2 << endl;
719
720 #if 1
721     String s3 = str2; //拷贝构造函数
722     cout << s3 << endl;
723     s3[3] = 'L';
724     cout << s3 << endl;
725     cout << str2 << endl;
726 #endif
727
728 }
729 int main() {
730     f();
731 }
732 #endif
733
734 #if 1
735 /*类
736    模拟vector<int>的类Vector
737 */
738 #include <iostream>
739 #include <string>
740 using namespace std;
741
```

```
742 class student {
743     string name;
744     double score;
745 public:
746     student(string n="no", double s=0) {
747         name = n; score = s;
748     }
749     friend ostream& operator<<(ostream &o, student s);
750 };
751
752 ostream& operator<<(ostream &o, student s) {
753     cout << s.name << "," << s.score << endl;
754     return o;
755 }
756
757 //类模板
758 template<typename T>
759 class Vector {
760     T *data;
761     int capacity;
762     int n;
763 public:
764     Vector(int cap=3) {
765         data = new T[cap];
766         if (data == 0) {
767             cap = 0; n = 0;
768             return;
769         }
770         capacity = cap;
771         n = 0;
772     }
773     void push_back(T e) {
774         if (n == capacity) {//空间已经满
775             cout << "增加容量！\n";
776             T *p = new T[2 * capacity];
777             if (p) {
778                 for (int i = 0; i < n; i++)
779                     p[i] = data[i];
780                 delete[] data;
781                 data = p;
782                 capacity = 2*capacity;
783             }
784             else {
785                 return;
786             }
787         }
788         data[n] = e;
789         n++;
790     }
791     T operator[](int i) const{
792         if (i < 0 || i >= n) throw "下标非法!";
793         return data[i];
794     }
```

```
795     int size() {
796         return n;
797     }
798 };
799 int main() {
800     Vector<student> v;
801     v.push_back(student("Li",45.7));
802     v.push_back(student("Wang", 45.7));
803     v.push_back(student("zhao", 45.7));
804 
805     for (int i = 0; i < v.size(); i++)
806         cout << v[i] ;
807     cout << endl;
808 
809     v.push_back(student("zhang", 45.7));
810     v.push_back(student("Liu", 45.7));
811     for (int i = 0; i < v.size(); i++)
812         cout << v[i];
813     cout << endl;
814 
815 #if 0
816 #if 1
817     Vector<int> v;
818     v.push_back(3);
819     v.push_back(4);
820     v.push_back(5);
821 
822     for(int  i = 0 ; i<v.size();i++)
823         cout<<v[i]<<'\t';
824     cout << endl;
825 
826     v.push_back(6);
827     v.push_back(7);
828     for (int i = 0; i < v.size(); i++)
829         cout << v[i] << '\t';
830     cout << endl;
831 #else
832     Vector<string> v;
833     v.push_back("hello");
834     v.push_back("world");
835     v.push_back("sdfasdf");
836 
837     for (int i = 0; i < v.size(); i++)
838         cout << v[i] << '\t';
839     cout << endl;
840 
841     v.push_back("ggg");
842     v.push_back("hhh");
843     for (int i = 0; i < v.size(); i++)
844         cout << v[i] << '\t';
845     cout << endl;
846 #endif
847 #endif
```

```
848
849 }
850 #endif
```