```
/*
//===========“从C到C++快速入门(2019版)”=====================
// youtube : https://www.youtube.com/watch?v=gZqO-gPArOo&list=PLBijWKRKPQMLrjj9yT7TWW9Bc23l-aBLx
//B站(bilibili.com): 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

}
```

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

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

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

```
212 using namespace std;
213 int add(int x, int y = 2) {
214     return x + y ;
215 }
216 double add(double x, double y = 2.0) {
217     return x + y;
218 }
219 int main() {
220     cout << add(5,3) << endl;
221     cout << add(5.3, 7.8) << endl;
222     cout << add((double)5, 7.8) << endl;//歧义性
223 }
224 #endif
225
226 #if 0
227 #include <iostream>
228 using namespace std;
229 int add(int x, int y) {
230     return x + y;
231 }
232 double add(double x, double y ) {
233     return x + y;
234 }
235 int main() {
236     cout << add(5, 3) << endl;
237     cout << add(5.3, 7.8) << endl;
238 //  cout << add("hello", "world") << endl;
239 }
240 #endif
241
242 #if 0
243 #include <iostream>
244 #include <string>
245 using namespace std;
246
247 template<typename T>
248 T add(T x, T y) {
249     return x + y;
250 }
251 int main() {
252 #if 0
253     cout << add<int>(5, 3) << endl;
254     cout << add<double>(5.3, 7.8) << endl;
255     cout << add<int>(4, 6) << endl;
256     cout << add<string>("hello", "world") << endl;
257 #else
258     cout << add(5, 3) << endl;
259     cout << add(5.3, 7.8) << endl;
260     cout << add((double)5, 7.8) << endl; //歧义性
261 #endif
262 }
263 #endif
264
```

```
265
266 #if 0
267 #include <iostream>
268 #include <string>
269 using namespace std;
270 int main() {
271     string s = "hello", s2("world");
272     //访问运算符.
273     cout << s.size() << endl;
274     string s3 = s.substr(1, 3);
275     cout << s3<< endl;
276
277     string s4 = s + " " + s2;
278     cout << s4 << endl; //"hello world"
279
280     s4[0] = 'H';
281     s4[6] = 'X';
282     cout << s4 << endl;
283
284     int pos = s4.find("orl");
285     cout << pos << endl;
286     s4.insert(3, "ABCDE");
287     cout << s4 << endl;
288
289     for (int i = 0; i < s4.size(); i++)
290         cout << s4[i] << "-";
291     cout << "\n";
292
293 }
294 #endif
295
296 #if 0
297 #include <iostream>
298 using std::cout;
299 int main() {
300     int arr[] = { 10,20,30,40 }; //大小固定，以后不能添加更多int值
301     for (int i = 0; i < 4; i++)
302         cout << arr[i] << '\t';
303     cout << '\n';
304 }
305 #endif
306
307 #if 0
308 #include <iostream>
309 #include <vector>
310 using namespace std;
311 int main() {
312     vector<int> v = { 7, 5, 16, 8 };
313     //push_back()，最后添加一个元素
314     v.push_back(25);
315     v.push_back(13);
316
317     //成员函数size()、下标运算符[]
```

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

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

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

```
477
478 #if 0
479 /*
480     this指针: 成员函数实际上隐含一个this指针。
481 */
482 #include <iostream>
483 #include <string>
484 using namespace std;
485
486 struct student {
487     string name;
488     double score;
489     void print() {
490         cout << this->name << " " << this->score << endl;
491     }
492 };
493 int main() {
494     student stu;
495     stu.name = "Li Ping";
496     stu.score = 78.5;
497     stu.print();   // print(&stu);
498 }
499 #endif
500
501
502
503 #if 0
504 /*
505     struct和class区别：
506     struct里的成员默认是public(公开的)
507     class里的成员默认是private(私有的)
508 */
509 #include <iostream>
510 #include <string>
511 using namespace std;
512
513 class student{
514 public: //接口
515     void print() {
516         cout << this->name << " " << this->score << endl;
517     }
518     string get_name() { return name; }
519     double get_score() { return score; }
520     void set_name(string n) { name = n; }
521     void set_score(double s) { score = s; }
522 private:
523     string name;
524     double score;
525 };
526 int main() {
527     student stu;
528
529 //  stu.name = "Li Ping";
```

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

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

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

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

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

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

```
847 #endif
848 #endif
849
850 }
851 #endif
```