# 1

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  typedef long long ll;  class xingzhuang  {  public:      string str;      double s;      xingzhuang()      {          str = "NOTHING";          s = 0;      }      virtual void show() = 0;  };  class Rectangle : public xingzhuang  {  public:      Rectangle(double ss)      {          s = ss;          str = "Rectangle";      }      void show()      {          cout << str << ": " << s << endl;      }  };  class Circle : public xingzhuang  {  public:      Circle(double ss)      {          s = ss;          str = "Circle";      }      void show()      {          cout << str << ": " << s << endl;      }  };  class Triangle : public xingzhuang  {  public:      Triangle(double ss)      {          s = ss;          str = "Triangle";      }      void show()      {          cout << str << ": " << s << endl;      }  };  int main()  {      ios::sync\_with\_stdio(false);      cin.tie(0);      cout.tie(0);      int n;      cin >> n;      xingzhuang \*arr[550];      for (int i = 0; i < n; i++)      {          char option;          cin >> option;          if (option == 'C')          {              double r;              cin >> r;              double s = r \* r \* 3.14;              arr[i] = new Circle(s);          }          else if (option == 'R')          {              double a, b;              cin >> a >> b;              double s;              s = b \* a;              arr[i] = new Rectangle(s);          }          else if (option == 'T')          {              double a, b, c;              cin >> a >> b >> c;              double p = (a + b + c) / 2;              double s = sqrt(p \* (p - a) \* (p - b) \* (p - c));              arr[i] = new Triangle(s);          }      }      for (int i = 0; i < n - 1; i++)      {          for (int j = i + 1; j < n; j++)          {              if (arr[i]->s < arr[j]->s)              {                  string stemp = arr[i]->str;                  double temp = arr[i]->s;                  arr[i]->s = arr[j]->s;                  arr[i]->str = arr[j]->str;                  arr[j]->s = temp;                  arr[j]->str = stemp;              }          }      }      for (int i = 0; i < n; i++)      {          arr[i]->show();      }      return 0;  } |
|  |
|  |

# 2

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  typedef long long ll;  class yuanGong  {  protected:      int id;      string name;      double money;  public:      yuanGong(int idd, string namee) : id(idd), name(namee), money(0) {}      virtual void calculate() = 0;      void show()      {          cout << "id " << id << " name " << name << " money " << money << endl;      }  };  class jingLi : virtual public yuanGong  {  public:      jingLi(int idd, string namee) : yuanGong(idd, namee) {}      void calculate()      {          money = 8000;      }  };  class jiShuRenRuan : virtual public yuanGong  {  protected:      int hour;  public:      jiShuRenRuan(int id, string name, int hour) : yuanGong(id, name), hour(hour) {}      void calculate()      {          money = hour \* 100;      }  };  class xiaoShouYuan : virtual public yuanGong  {  protected:      double xiaoshouLiang;  public:      xiaoShouYuan(int id, string name, double xiao) : yuanGong(id, name), xiaoshouLiang(xiao) {}      void calculate()      {          money = xiaoshouLiang / 100.0 \* 4.0;      }  };  class xiaoshoujingli : public xiaoShouYuan, public jingLi  {  public:      xiaoshoujingli(int id, string name, double xiao) : yuanGong(id, name), xiaoShouYuan(id, name, xiao), jingLi(id, name) {}      void calculate()      {          money = xiaoshouLiang / 1000.0 \* 5.0 + 5000;      }  };  int main()  {      ios::sync\_with\_stdio(false);      cin.tie(0);      cout.tie(0);      cout << "月工资计算办法是：经理拿固定月薪 8000 元，兼职技术人员按每小时 100 元领取月薪，销售员按当月销售额的 4 %提成，销售经理既拿固定月工资也领取销售提成，固定月工资为 5000 元，销售提成为所管辖部门当月销售总额的 5‰ " << endl;      cout << "技术人员干了1000h,销售卖了100000,销售经理卖了1000" << endl;      yuanGong \*arr[1000];      arr[0] = new jingLi(123, "woShiJingLi");      arr[1] = new jiShuRenRuan(666, "JishuRenYuan", 1000);      arr[2] = new xiaoShouYuan(114514, "xiaoShouRenYuan", 100000);      arr[3] = new xiaoshoujingli(23333, "AAAxiaoShouJingLi", 1000);      for (int i = 0; i < 4; i++)      {          arr[i]->calculate();      }      for (int i = 0; i < 4; i++)      {          arr[i]->show();      }      return 0;  } |
|  |
|  |

# 3

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  typedef long long ll;  class Venue // 运动场馆  {  public:      int id;      string name;      virtual void show() = 0;  };  class Site : public Venue // 场地  {  public:      double area;      Site(double ss, int id2, string name2) // 输入面积,id,name      {          area = ss;          name = name2;          id = id2;      }      void show()      {          cout << " id是: " << id << " 名字是: " << name << " 一个site面积是: " << area << endl;      }  };  class Gymnasium : public Venue // 运动馆  {  public:      bool airCondition;      Gymnasium(bool wenhao, int id2, string name2) // 有空调吗      {          airCondition = wenhao;          name = name2;          id = id2;      }      void show()      {          cout << " id是: " << id << " 名字是: " << name << " 是一个Gymnasium ";          if (airCondition)              cout << " 有空调 " << endl;          else              cout << " 没有空调 " << endl;      }  };  class VenueList  {      set<Venue \*> q;  public:      void add(Venue \*a)      {          q.insert(a);      }      void del(Venue \*a)      {          q.erase(a);      }      void show()      {          for (auto it = q.begin(); it != q.end(); it++)          {              (\*it)->show();          }      }  };  class Record  {  public:      string person;      string time;      string phone;      Venue \*a;      Record(string pe, string ti, string ph, Venue \*temp)      {          person = pe;          time = ti;          phone = ph;          a = temp;      }      Record(){}      void show()      {          cout << "姓名: " << person << " 时间 " << time << " 电话号码 " << phone << " 去了 " << a->name << endl;      }  };  class RecordList : public Record  {      set<Record \*> p;  public:      RecordList() {}      void add(Record \*a)      {          p.insert(a);      }      void rem(Record \*a)      {          p.erase(a);      }      void show()      {          for (auto it = p.begin(); it != p.end(); it++)          {              (\*it)->show();          }      }  };  int main()  {      VenueList venueList;      // 创建场地和运动馆      Site site1(100.0, 1, "Site 1");      Gymnasium gym1(true, 2, "Gymnasium 1");      Gymnasium gym2(false, 3, "Gymnasium 2");      // 添加场地和运动馆到场馆列表      venueList.add(&site1);      venueList.add(&gym1);      venueList.add(&gym2);      venueList.show();      cout << " 由于电力不足,学校关闭有空调的体育馆" << endl;      venueList.del(&gym1);      venueList.show();      Record a("liujunfeng", "2024.4.9.17.52", "114514", &gym2);      Record b("Uchaoman", "1920.1.1.1.0", "911", &site1);      RecordList recordlist;      recordlist.add(&a);      recordlist.add(&b);      recordlist.show();      cout << " liujunfeng累了" << endl;      recordlist.rem(&a);      recordlist.show();      return 0;  } |
|  |
|  |

# 4

|  |
| --- |
| #include <iostream>  using namespace std;  typedef long long ll;  template <class T>  void sort(T a[], int n)  {      for (int i = 1; i < n; i++)      {          T temp = a[i];          int j = i - 1;          while (j >= 0 && a[j] > temp)          {              a[j + 1] = a[j];              j--;          }          a[j + 1] = temp;      }  }  int main()  {      ios::sync\_with\_stdio(false);      cin.tie(0);      cout.tie(0);      int arr[] = {5, 4, 9, 0, 4, 1, 6, 8, 2, 3, 4, 7};      char str[] = {'z', 'g', 'a', 'b', 'q', 'p', 'b', 'd', 'n'};      int n1 = sizeof(arr) / sizeof(int);      int n2 = sizeof(str) / sizeof(char);      for (int i = 0; i < n1; i++)      {          cout << arr[i] << ' ';      }      cout << endl;      for (int i = 0; i < n2; i++)      {          cout << str[i] << ' ';      }      cout << endl;      sort(arr, n1);      for (int i = 0; i < n1; i++)      {          cout << arr[i] << ' ';      }      cout << endl;      sort(str, n2);      for (int i = 0; i < n2; i++)      {          cout << str[i] << ' ';      }      return 0;  } |
|  |
|  |