|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  class Time  {  private:      int hour = 0;      int minute = 0;      int second = 0;  public:      void Set(int h = 0, int m = 0, int s = 0) // 默认参数设置时间      {          hour = h;          minute = m;          second = s;      }      void Set(int h, int m) // 重载函数设置时间，和默认参数二选一即可      {          Set(h, m, 0);      }      void Set(int s) // 重载函数设置时间，和默认参数二选一即可      {          Set(s, 0, 0);      }      void Tick(); // +1s      void Show() // 输出时间，采用24小时制，参考H:M:S      {          cout << hour << ':' << minute << ":" << second  << endl;      }  };  void Time::Tick()  {      second++;      minute += (second) / 60;      second %= 60;      hour += minute / 60;      minute %= 60;      hour %= 24;  }  int main()  {      Time time;      time.Set(13, 1, 59); // 采用24小时制      time.Show();         // 输出13:1:59      time.Tick();      time.Show(); // 输出13:2:0      time.Set(13, 59, 59);      time.Tick();      time.Show(); // 输出14:0:0      time.Set(23, 59, 59);      time.Tick();      time.Show(); // 输出0:0:0      return 0;  } |
|  |

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  class User  {  private:      unsigned int id = 0; // id 非负      char password[6];    // 6为密码，默认000000  public:      User(unsigned int iiid) // 根据id构造User对象      {          if (iiid >= 0)              id = iiid;          else              cout << "id 非负" << endl;          for (int i = 0; i < 6; i++)              password[i] = '0';      }      void PrinId() // 输出id      {          cout << id << endl;      }      void SetPassword(char \*oldPassword, char \*newPassword) // 设置密码，比较新旧密码，提示操作是否成功      {          int right = 1;          for (int i = 0; i < 6; i++)          {              if (password[i] != oldPassword[i])                  right \*= 0;          }          for (int i = 0; i < 6; i++)          {              if (newPassword[i] >= '0' && newPassword[i] <= '9')                  right \*= 1;              else                  right \*= 0;          }          if (strlen(newPassword) > 6)              right \*= 0;          if (right)          {              for (int i = 0; i < 6; i++)              {                  password[i] = newPassword[i];              }              cout << "密码设置成功" << endl;          }          else              cout << "设置密码失败" << endl;      }  };  int main()  {      User user(123456);      user.PrinId();                         // 输出123456      user.SetPassword("000000", "11111A");  // 输出密码设置失败      user.SetPassword("000000", "1111111"); // 输出密码设置失败      user.SetPassword("0", "111111");       // 输出密码设置失败      user.SetPassword("100000", "111111");  // 输出密码设置失败      user.SetPassword("000000", "111111");  // 输出密码设置成功      return 0;  } |
|  |

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  typedef long long ll;  int gcd(int x, int y)  {      if (x == 0)          return 1;      while (y ^= x ^= y ^= x %= y)          ;      return x;  }  class Rational  {  public:      Rational(int nn = 1, int mm = 1) : n(nn), m(mm)      {          if (mm == 0)              cout << "分母不能为0" << endl;      }      Rational add(Rational &A)      {          Rational result(n \* A.m + A.n \* m, A.m \* m);          result.simple();          return result;      }      Rational sub(Rational &A)      {          Rational result(n \* A.m - A.n \* m, A.m \* m);          result.simple();          return result;      }      Rational mul(const Rational &A)      {          Rational result(n \* A.n, A.m \* m);          result.simple();          return result;      }      Rational div(const Rational &A)      {          if (A.n == 0)          {              cout << "除数不能为0" << endl;          }          else          {              Rational result(n \* A.m, m \* A.n);              result.simple();              return result;          }      }      void print()      {          cout << n << "/" << m << endl;      }  private:      void simple()      {          int maxGcd = gcd(n, m);          n /= maxGcd;          m /= maxGcd;          n = n \* m / abs(m);          m = abs(m);      }      int n;      int m;  };  int main()  {      Rational a(1, 2), b(2, 3);      Rational result = a.add(b);      result.print(); // 7/6      result = a.sub(b);      result.print(); // -1/6      result = a.mul(b);      result.print(); // 1/3      result = a.div(b);      result.print(); // 3/4      Rational c(0, 1);      result = a.mul(c); // 乘0      result.print();      result = a.div(c);        // 不能除以0      Rational test\_zero(1, 0); // 分母不能为0      return 0;  } |
|  |

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  typedef long long ll;  int isRun(int year)  {      if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0))          return 1;      else          return 0;  }  class Date  {  public:      Date(int y, int m, int d)      {          if (is\_valid(y, m, d))          {              year = y;              month = m;              day = d;          }          else          {              cout << "输入日期错误,已帮您自动设置为2024年1月1号" << endl;              year = 2024;              month = 1;              day = 1;          }      }      Date()      {          cout << "由于你没输入日期,已帮您自动设置为2024年1月1号" << endl;          year = 2024;          month = 1;          day = 1;      }      bool set(int y, int m, int d)      {          year = y;          month = m;          day = d;          return true;      }      void print()      {          cout << month << "/" << day << "/" << year << endl;      }      int day\_of\_year()      {          int days[] = {0, 31, 28 + isRun(year), 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};          int i = 0;          int DiJiTiannn = 0;          for (i = 1; i < month; i++)          {              DiJiTiannn += days[i];          }          DiJiTiannn += day;          return DiJiTiannn;      }      int diff(const Date &A)      {          int First = Date::day\_of\_year();          int days[] = {0, 31, 28 + isRun(A.year), 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};          int i = 0;          int DiJiTiannn = 0;          for (i = 1; i < A.month; i++)          {              DiJiTiannn += days[i];          }          DiJiTiannn += A.day;          int Second = DiJiTiannn;          for (int i = 1999; i < year; i++)          {              First += 365;              First += isRun(i);          }          for (int i = 1999; i < A.year; i++)          {              Second += 365;              Second += isRun(i);          }          return First - Second;      }  private:      bool is\_valid(int y, int m, int d)      {          if (y < 1 || m < 1 || m > 12 || d < 1)              return false;          int days[] = {0, 31, 28 + isRun(y), 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};          if (d > days[m])              return false;          return true;      }      int year;      int month;      int day;  };  int main()  {      Date invalid\_date(2020, 2, 30);      invalid\_date.print(); // 输出默认日期      Date default\_date;      default\_date.print(); // 输出默认日期      Date a(2024, 3, 11);      a.print(); // 输出 3/11/2024      a.set(2019, 2, 29);      a.print(); // 输出默认日期      a.set(2024, 2, 29);      a.print(); // 输出 2/29/2024      a.set(2024, 3, 11);      cout << a.day\_of\_year() << endl; // 输出 71      Date b(2024, 5, 1);      cout << a.diff(b) << endl; // 输出 -51      b.set(2023, 12, 25);      cout << a.diff(b) << endl; // 输出 77      cout << a.diff(a) << endl; // 输出 0  } |
|  |

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  typedef long long ll;  class Array  {  public:      Array(int n) // n个元素. 所有元素初始为 0      {          data = new int[n];          size = n;          for (int i = 0; i < n; i++)          {              data[i] = 0;          }      }      Array(int \*p, int n) // 用一个整型原生C数组初始化      {          data = new int[n];          size = n;          for (int i = 0; i < n; i++)          {              data[i] = p[i];          }      }      Array(const Array &other) // 拷贝构造函数      {          data = new int[other.size];          size = other.size;          for (int i = 0; i < size; i++)          {              data[i] = other.data[i];          }      }      ~Array() // 析构函数      {          delete[] data;      }      void print() // 输出所有元素      {          for (int i = 0; i < size; i++)          {              cout << data[i] << ' ';          }          cout << endl;      }      void input() // 键盘输入所有元素      {          for (int i = 0; i < size; i++)          {              cin >> data[i];          }      }      void set(int index, int value) // 将index号元素的值设置为value      {          if (index >= 0 && index < size)              data[index] = value;          else              cout << "数组越界访问" << endl;      }  private:      int \*data;      int size; // number of elements  };  int main()  {      Array arr1(4);      arr1.input();   // 输入1 2 3 4      arr1.print();   // 输出1 2 3 4      arr1.set(5, 6); // 输出数组越界！      arr1.print();   // 输出1 2 3 4      arr1.set(2, 6);      arr1.print();      // 输出1 2 6 4      Array arr2 = arr1; // 调用拷贝构造函数对arr2初始化      arr1.print();      // 输出1 2 6 4      int arr[6] = {1, 2, 3, 4, 5, 6};      Array arr3(arr, 6);      arr3.print(); // 输出1 2 3 4 5 6      Array arr4(arr + 2, 4);      arr4.print(); // 输出3,4,5,6      return 0;  } |
|  |