**BÀI TẬP**

**Bài 1: Búp bê Nga**

|  |
| --- |
| #include <iostream>  #include <vector>  #include <queue>  void bubbleSort(std::vector<int>& arr) {    int n = arr.size();    for (int i = 0; i < n-1; i++) {      for (int j = 0; j < n-i-1; j++) {        if (arr[j] < arr[j+1]) {          int temp = arr[j];          arr[j] = arr[j+1];          arr[j+1] = temp;        }      }    }  }  int main() {    int n;    std::cin >> n;    std::vector<int> heights(n);    for (int i = 0; i < n; ++i) {      std::cin >> heights[i];    }    bubbleSort(heights);    std::queue<int> q;    for (int height : heights) {      if (!q.empty() && q.front() - height >= 3) {        q.pop();      }      q.push(height);    }    std::cout << q.size() << std::endl;    return 0;  } |

**Bài 2: Quản lý sinh viên**

|  |
| --- |
| #include <iostream>  #include <string>  #include <list>  class SinhVien {    private:      int maSo;      std::string ten;      int namSinh;      std::string gioiTinh;      double diemToan;      double diemLy;      double diemHoa;      double diemTong;    public:      SinhVien(int maSo, std::string ten, int namSinh, std::string gioiTinh, double diemToan, double diemLy, double diemHoa)        : maSo(maSo), ten(ten), namSinh(namSinh), gioiTinh(gioiTinh), diemToan(diemToan), diemLy(diemLy), diemHoa(diemHoa) {        diemTong = diemToan + diemLy + diemHoa;      }      friend std::istream& operator>>(std::istream &in, SinhVien &sinhVien) {        std::cout << "Nhap ma so: ";        in >> sinhVien.maSo;        std::cout << "Nhap ten: ";        in.ignore();        std::getline(in, sinhVien.ten);        std::cout << "Nhap nam sinh: ";        in >> sinhVien.namSinh;        std::cout << "Nhap gioi tinh: ";        in >> sinhVien.gioiTinh;        std::cout << "Nhap diem toan: ";        in >> sinhVien.diemToan;        std::cout << "Nhap diem ly: ";        in >> sinhVien.diemLy;        std::cout << "Nhap diem hoa: ";        in >> sinhVien.diemHoa;        sinhVien.diemTong = sinhVien.diemToan + sinhVien.diemLy + sinhVien.diemHoa;        return in;      }      friend std::ostream& operator<<(std::ostream &out, const SinhVien &sinhVien) {        out << "Ma so: " << sinhVien.maSo << "\n";        out << "Ten: " << sinhVien.ten << "\n";        out << "Nam sinh: " << sinhVien.namSinh << "\n";        out << "Gioi tinh: " << sinhVien.gioiTinh << "\n";        out << "Diem toan: " << sinhVien.diemToan << "\n";        out << "Diem ly: " << sinhVien.diemLy << "\n";        out << "Diem hoa: " << sinhVien.diemHoa << "\n";        out << "Diem tong: " << sinhVien.diemTong << "\n";        return out;      }      int getMaSo() const { return maSo; }      double getDiemTong() const { return diemTong; }  };  class QuanLySinhVien {    private:      std::list<SinhVien> sinhViens;    public:      void themSinhVien(const SinhVien &sinhVien) {        sinhViens.push\_back(sinhVien);      }      SinhVien\* timSinhVienTheoMaSo(int maSo) {        for (auto &sinhVien : sinhViens) {          if (sinhVien.getMaSo() == maSo) {            return &sinhVien;          }        }        return nullptr;      }      void xoaSinhVienTheoMaSo(int maSo) {        sinhViens.remove\_if([maSo](const SinhVien &sinhVien) { return sinhVien.getMaSo() == maSo; });      }      void hienThiTatCaSinhVien() const {        for (const auto &sinhVien : sinhViens) {          std::cout << sinhVien << "\n";        }      }  };  int main() {    QuanLySinhVien quanLy;    SinhVien sinhVien1(1, "Nguyen Van A", 2000, "Nam", 8.5, 7.0, 9.0);    SinhVien sinhVien2(2, "Tran Thi B", 2001, "Nu", 6.0, 8.0, 7.5);    quanLy.themSinhVien(sinhVien1);    quanLy.themSinhVien(sinhVien2);    std::cout << "Tat ca sinh vien:\n";    quanLy.hienThiTatCaSinhVien();    SinhVien\* sinhVienTimThay = quanLy.timSinhVienTheoMaSo(1);    if (sinhVienTimThay) {      std::cout << "Sinh vien tim thay:\n" << \*sinhVienTimThay;    } else {      std::cout << "Khong tim thay sinh vien.\n";    }    quanLy.xoaSinhVienTheoMaSo(1);    std::cout << "Tat ca sinh vien sau khi xoa:\n";    quanLy.hienThiTatCaSinhVien();    return 0;  } |

**Bài 3: Quản lý laptop**

|  |
| --- |
| #include <iostream>  #include <vector>  #include <string>  class Laptop {    private:      std::string nhaSanXuat;      std::string model;      std::string cauHinh;      double gia;      int soLuong;    public:      Laptop(std::string nhaSanXuat, std::string model, std::string cauHinh, double gia, int soLuong)        : nhaSanXuat(nhaSanXuat), model(model), cauHinh(cauHinh), gia(gia), soLuong(soLuong) {}      friend std::istream& operator>>(std::istream &in, Laptop &laptop) {        std::cout << "Nhap nha san xuat: ";        in >> laptop.nhaSanXuat;        std::cout << "Nhap model: ";        in >> laptop.model;        std::cout << "Nhap cau hinh: ";        in >> laptop.cauHinh;        std::cout << "Nhap gia: ";        in >> laptop.gia;        std::cout << "Nhap so luong: ";        in >> laptop.soLuong;        return in;      }      friend std::ostream& operator<<(std::ostream &out, const Laptop &laptop) {        out << "Nha san xuat: " << laptop.nhaSanXuat << "\n";        out << "Model: " << laptop.model << "\n";        out << "Cau hinh: " << laptop.cauHinh << "\n";        out << "Gia: " << laptop.gia << "\n";        out << "So luong: " << laptop.soLuong << "\n";        return out;      }      std::string getNhaSanXuat() const { return nhaSanXuat; }      std::string getModel() const { return model; }      std::string getCauHinh() const { return cauHinh; }      double getGia() const { return gia; }      int getSoLuong() const { return soLuong; }  };  class QuanLyLaptop {    private:      std::vector<Laptop> laptops;    public:      void themLaptop(const Laptop &laptop) {        laptops.push\_back(laptop);      }      double tinhTongGiaTriTheoNhaSanXuat(const std::string &nhaSanXuat) const {        double tong = 0;        for (const auto &laptop : laptops) {          if (laptop.getNhaSanXuat() == nhaSanXuat) {            tong += laptop.getGia() \* laptop.getSoLuong();          }        }        return tong;      }      void lietKeLaptopTheoKichThuocManHinh(const std::string &kichThuocManHinh) const {        for (const auto &laptop : laptops) {          if (laptop.getCauHinh().find(kichThuocManHinh) != std::string::npos) {            std::cout << laptop;          }        }      }      Laptop\* timKiemLaptopTheoModel(const std::string &model) {        int left = 0;        int right = laptops.size() - 1;        while (left <= right) {          int mid = left + (right - left) / 2;          std::string midModel = laptops[mid].getModel();          if (midModel == model) {            return &laptops[mid];          } else if (midModel < model) {            left = mid + 1;          } else {            right = mid - 1;          }        }        return nullptr;      }      void hienThiTatCaLaptop() const {        for (const auto &laptop : laptops) {          std::cout << laptop;        }      }  };  int main() {    QuanLyLaptop quanLy;    Laptop laptop1("Dell", "XPS13", "Intel i7, 16GB RAM, 512GB SSD, 13 inch", 1200, 10);    Laptop laptop2("HP", "Spectre x360", "Intel i5, 8GB RAM, 256GB SSD, 13 inch", 1000, 15);    quanLy.themLaptop(laptop1);    quanLy.themLaptop(laptop2);    std::cout << "Tong gia tri cua cac laptop Dell: $" << quanLy.tinhTongGiaTriTheoNhaSanXuat("Dell") << "\n";    std::cout << "Cac laptop co kich thuoc man hinh 13 inch:\n";    quanLy.lietKeLaptopTheoKichThuocManHinh("13 inch");    Laptop\* laptopTimThay = quanLy.timKiemLaptopTheoModel("Spectre x360");    if (laptopTimThay) {      std::cout << "Tim thay laptop:\n" << \*laptopTimThay;    } else {      std::cout << "Khong tim thay laptop.\n";    }    std::cout << "Tat ca cac Laptop:\n";    quanLy.hienThiTatCaLaptop();    return 0;  } |

**Bài 4: Quản lý nhân viên**

|  |
| --- |
| #include <iostream>  #include <list>  using namespace std;  class NhanVien {  private:    int maNV;    string hoTen;    int namSinh;    double luong;  public:    NhanVien() {}    NhanVien(int maNV, string hoTen, int namSinh, double luong) {      this->maNV = maNV;      this->hoTen = hoTen;      this->namSinh = namSinh;      this->luong = luong;    }    friend istream& operator>>(istream& is, NhanVien& nv) {      cout << "Nhap ma nhan vien: ";      is >> nv.maNV;      cout << "Nhap ho ten: ";      is.ignore();      getline(is, nv.hoTen);      cout << "Nhap nam sinh: ";      is >> nv.namSinh;      cout << "Nhap luong: ";      is >> nv.luong;      return is;    }    friend ostream& operator<<(ostream& os, NhanVien& nv) {      os << "Ma nhan vien: " << nv.maNV << endl;      os << "Ho ten: " << nv.hoTen << endl;      os << "Nam sinh: " << nv.namSinh << endl;      os << "Luong: " << nv.luong << endl;      return os;    }    int getMaNV() const {      return maNV;    }  };  class QuanLyNhanVien {  private:    list<NhanVien> dsNhanVien;  public:    void themNhanVien(NhanVien nv) {      dsNhanVien.push\_back(nv);    }    void xoaNhanVienTheoMa(int maNV) {      for (auto it = dsNhanVien.begin(); it != dsNhanVien.end(); ++it) {        if ((\*it).getMaNV() == maNV) {          it = dsNhanVien.erase(it);          break;        }      }    }    void inDanhSachNhanVien() {      for (auto nv : dsNhanVien) {        cout << nv << endl;      }    }    NhanVien timKiemNhanVienTheoMa(int maNV) {      for (auto nv : dsNhanVien) {        if (nv.getMaNV() == maNV) {          return nv;        }      }      return NhanVien();    }  };  int main() {    QuanLyNhanVien qlnv;    NhanVien nv1(1, "Nguyen Van A", 1990, 1000);    NhanVien nv2(2, "Nguyen Van B", 1991, 2000);    NhanVien nv3(3, "Nguyen Van C", 1992, 3000);    qlnv.themNhanVien(nv1);    qlnv.themNhanVien(nv2);    qlnv.themNhanVien(nv3);    qlnv.inDanhSachNhanVien();    cout << "Nhap ma nhan vien can xoa: ";    int maNV;    cin >> maNV;    qlnv.xoaNhanVienTheoMa(maNV);    qlnv.inDanhSachNhanVien();    cout << "Nhap ma nhan vien can tim: ";    cin >> maNV;    NhanVien nv = qlnv.timKiemNhanVienTheoMa(maNV);    if (nv.getMaNV() != 0) {      cout << nv << endl;    } else {      cout << "Khong tim thay nhan vien" << endl;    }    return 0;  } |