

TRƯỜNG ĐẠI HỌC NGOẠI NGỮ - TIN HỌC
THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN

ĐỀ THI KẾT THÚC HỌC PHẦN
BẠC ĐẠI HỌC – HỆ CHÍNH QUY
Học kỳ 2, Năm học 2019 – 2020
Khóa 2019

ĐỀ CHÍNH THỨC
(Đề có 3 trang)

Môn: Kỹ thuật lập trình
Thời gian làm bài: 90 phút, không kể thời gian phát đề
(SV được sử dụng tài liệu)

Câu 1. (04 điểm) Cho dãy số nguyên $a = (a_1, a_2, \dots, a_n)$ ($n \leq 1000$). Hãy tìm tất cả các số nguyên tố có trong dãy số nguyên a và sắp xếp dãy các số nguyên tố vừa tìm được (nếu có) bằng thuật toán Interchange Sort.

Input

- Dòng đầu tiên chứa số nguyên n
- Dòng thứ hai chứa n số nguyên của dãy a

Yêu cầu: Hãy viết các hàm sau để giải quyết bài toán trên:

a) Viết hàm nhập dãy số nguyên a theo mô tả trong phần Input:

```
void Input(out int[] a) {...}
```

b) Viết hàm kiểm tra số nguyên x có phải là số nguyên tố không:

```
bool IsPrime(int x) {...}
```

c) Viết hàm tìm và trả về danh sách tất cả các số nguyên tố có trong dãy a :

```
List<int> FindAllPrime(int[] a) {...}
```

d) Viết hàm sắp xếp danh sách số nguyên bằng thuật toán InterchangeSort (không sử dụng phương thức Sort của kiểu dữ liệu List):

```
void InterchangeSort(List<int> intList) {...}
```

```
class Program
{
    static void Input(out int[] a)
    {
        //TO DO
    }
    static bool IsPrime(int x)
    {
        //TO DO
    }
    static List<int> FindAllPrime(int[] a)
    {
        //TO DO
    }
    static void InterchangeSort(List<int> intList)
    {
        //TO DO
    }
    static void Main(string[] args)
    {
        int[] a;
        Input(out a);
        List<int> primeNumbers = FindAllPrime(a);
        if (primeNumbers.Count > 0)
            InterchangeSort(primeNumbers);
    }
}
```

Ghi chú:

- Anh/Chị viết lại class Program và cài đặt phần “TO DO” của 4 hàm Input IsPrime, FindAllPrime và InterChangeSort; không cần viết lại hàm Main().
- Hàm nào không làm được, vẫn có thể được sử dụng để cài đặt những hàm khác.

Câu 2. (06 điểm) Một công ty phần mềm đang có nhu cầu quản lý các nhân viên của công ty. Mỗi nhân viên bao gồm các thông tin cơ bản sau: Mã NV, Họ Tên NV, Hệ số lương. Nhân viên trong công ty sẽ được tính lương dựa vào mức Lương căn bản áp dụng chung cho toàn bộ nhân viên và Hệ số lương của nhân viên đó.

a) Định nghĩa lớp NhanVien: (04 điểm)

- Thuộc tính:

STT	Tên thuộc tính	Phạm vi truy cập	Kiểu dữ liệu	Mô tả
1	staffID	private	string	Mã NV
2	name	private	string	Họ Tên NV
3	payRate	private	double	Hệ số lương
4	baseRate	private	double	Mức Lương căn bản dùng chung cho toàn bộ nhân viên của công ty (thuộc tính static)

- Phương thức khởi tạo (Constructor):
 - Constructor mặc định (không tham số): staffID = “null”, name = “null”.
 - Constructor có 3 tham số là staffID, name, payRate.
- Phương thức:
 - Thiết lập mức Lương cơ bản cho toàn bộ công ty:
`static void SetBaseRate(double baseRate) {...}`
 - Nhập thông tin một nhân viên, bao gồm: Mã NV, Họ tên và Hệ số lương. Yêu cầu người dùng nhập Hệ số lương > 0, nếu nhập sai thì yêu cầu nhập lại:
`void Input() {...}`
 - Trả về chuỗi biểu diễn thông tin của một nhân viên có dạng: “NhanVien[MaNV = ???, HoTen = ???, HS Luong = ???]”
`string toString() {...}`
 - Tính lương cho 1 nhân viên, biết lương của nhân viên được tính theo công thức: $Lương = Lương cơ bản \times Hệ số lương$.
`double Pay() {...}`

b) Viết hàm **Main()** thực hiện chức năng sau: (02 điểm)

- Khai báo một đối tượng thuộc lớp **NhanVien**.
- Nhập thông tin, tính lương và xuất thông tin của nhân viên đó (bao gồm thông tin cơ bản và lương). Ví dụ:
NhanVien[MaNV = 123, HoTen = Khanh, HS Luong = 2.31] -
Luong: 10395000
- Khai báo 1 danh sách các đối tượng thuộc lớp **NhanVien**.
- Nhập thông tin cho danh sách nhân viên trên.

--- Hết ---