

Module/môn: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT	Số hiệu assignment: 1/1	% điểm: 60%
Người điều phối của FPT Polytechnic: NghiemN	Ngày ban hành: 7/01/2020	
Bài assignment này đòi hỏi sinh viên phải dùng khoảng 24h làm để hoàn thành		
Tương ứng với mục tiêu môn học: A, B,		

Gian lận là hình thức lấy bài làm của người khác và sử dụng như là mình làm ra. Hình thức đó bao gồm những hành động như: copy thông tin trực tiếp từ trang web hay sách mà không ghi rõ nguồn tham khảo trong tài liệu; gửi bài assignment làm chung như là thành quả cá nhân; copy bài assignment của các sinh viên khác cùng khóa hay khác khóa; ăn trộm hay mua bài assignment của ai đó và gửi lên như là sản phẩm mình làm ra. Những sinh viên bị nghi ngờ gian lận sẽ bị điều tra và nếu phát hiện là có gian lận thì sẽ phải chịu các mức phạt theo quy định của Nhà trường.

Mọi tài nguyên copy hay điều chế từ bất cứ nguồn nào (VD: Internet, sách) phải được đặt trong cặp dấu nháy kép và in nghiêng, với thông tin tham khảo đầy đủ về nguồn tài liệu.

Bài làm của bạn sẽ được đưa vào phần mềm kiểm tra gian lận. Mọi hình thức cố tình đánh lừa hệ thống phát hiện gian lận sẽ bị coi là Vi phạm quy định thi cử.

Quy định nộp bài assignment

- Một bản mềm kết quả bài làm assignment của bạn phải được upload trước nửa đêm (giờ địa phương) vào ngày hạn nộp. **Quá hạn nộp hệ thống sẽ khóa lại và sinh viên không còn quyền nộp bài.**
- Phiên bản upload lên cuối cùng sẽ được chấm điểm. Sinh viên có quyền upload đề file nhiều lần trước khi hết hạn nộp.
- Tất cả những file tài liệu văn bản phải để ở dạng file gốc chứ không file dạng file được xuất ra từ định dạng khác (ví dụ pdf được xuất từ doc). Không được gửi tài liệu văn bản dưới dạng ảnh chụp.
- Đối với bài assignment này bạn cũng phải đưa các bằng chứng hay sản phẩm khác vào trong file nén dạng zip.
- Kích thước file cần tuân thủ theo giới hạn trên hệ thống nộp bài (thông thường là <50M).

- Hãy đảm bảo các file được upload lên không bị nhiễm virus (**điều này có thể dẫn đến file bị hệ thống xóa mất**) và không đặt mật khẩu mở file. Nếu vi phạm những điều này, bài coi như chưa được nộp.
- Hãy chú ý xem thông báo sau khi upload để chắc chắn bài của bạn đã được nộp lên hệ thống chưa.
- Bạn không phải gửi lại file đề bài của assignment (file này).

Quy định đánh giá bài assignment

1. Sinh viên không có bài assignment trên hệ thống sẽ bị 0 điểm bài assignment.
2. Sau hạn nộp bài một tuần, sinh viên nộp muộn có quyền nộp đơn kiến nghị xin được chấp nhận gia hạn nộp. Hội đồng Nhà trường sẽ xét duyệt từng trường hợp. Nếu kiến nghị không được chấp nhận, bài giữ nguyên điểm 0. Nếu quá một tuần không có kiến nghị thì bài cũng sinh viên không nộp mặc nhiên nhận điểm 0.
3. Ngay cả trường hợp bài của sinh viên bị phát hiện gian lận sau khi có điểm, sinh viên sẽ không được công nhận bài đó và chịu mức kỷ luật như quy định của Nhà trường.

=====

Assignment

Xây dựng ứng dụng quản lý nhân sự

Mục tiêu cụ thể	Sau bài assignment này, sinh viên sẽ biết cách: <ul style="list-style-type: none">- Xây dựng cấu trúc dữ liệu vào ứng dụng thực tế.- Ứng dụng các giải thuật tìm kiếm vào bài toán thực tế.- Ứng dụng các giải thuật sắp xếp vào bài toán thực tế.
------------------------	--

Các công cụ cần có	<ul style="list-style-type: none"> - Dev c++ - MinGW GCC - Hoặc các công cụ phần mềm có hỗ trợ viết c/c++ (Netbean, visual – studio...)
Tham khảo	<ul style="list-style-type: none"> - https://vietjack.com/cau-truc-du-lieu-va-giai-thuat/ - https://nguyenvanhieu.vn/cau-truc-du-lieu/ - https://www.e.ptit.edu.vn/cau-truc-du-lieu-va-giai-thuat/

ĐẶT VẤN ĐỀ

Ứng dụng các cấu trúc dữ liệu đã được học và các giải thuật tìm kiếm, sắp xếp, sinh viên viết chương trình quản lý nhân sự cho một công ty mà bạn đang làm việc, mỗi nhân viên trong công ty gồm các thông tin sau: mã số, họ, tên, ngày sinh, nơi sinh, địa chỉ, ngày công tác, lương...

YÊU CẦU

Các công việc yêu cầu cụ thể như sau:

Y1 – Yêu cầu về cách trình bày nội dung code

1. Có chú thích rõ ràng về chức năng của từng hàm.
2. Thể hiện cấp cha-con khi viết các câu lệnh.

Y2 – Yêu cầu về chức năng của hệ thống

1. Menu quản lý nhân viên.

🔗 Khai báo thư viện

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#include<string.h>
#define sl 100
```

🔗 Khai báo Struct NhanVien, Date, DanhSach

```
struct Date
{
    int ngay;
    int thang;
    int nam;
};
typedef Date date;
struct NhanVien
{
    int msnv;
    char ho[20];
    char ten[20];
    date namsinh;
    char noisinh[100];
    char diachi[200];
    float luong;
    date ngayvao;
};
typedef NhanVien nhanvien;
struct DanhSach
{
    nhanvien a[51];
    int n;
};
typedef DanhSach danhsach;
```

✚ Khai báo các hàm cần sử dụng

```

void menu();
void nhapngaythang(date &ntn);
void nhap1nv(nhanvien &nv);
void xuat1nv(nhanvien nv);
void xuatngaythang(date ntn);
void nhapdanhsach(danhsach &ds);
void xuatdanhsach(danhsach ds);
int timten(danhsach ds,char *c);
void themnv(danhsach &ds,nhanvien x,int k);
int timMSNV(danhsach ds,int d);
void xoanv(danhsach ds,int d);
// ham sap xep danh
void sapxep(danhsach ds);
int main()
{
    menu();
    getch();
}

```

```

void menu()
{
    int ch;
    int d,k;
    danhsach ds;
    nhanvien x;
    nhapdanhsach(ds);
    xuatdanhsach(ds);
    printf("\n\t\t-----MENU-----");
    printf("\n\t1.Them vao 1 nhan vien");
    printf("\n\t2.Tim nhan vien theo ma so");
    printf("\n\t3.Tim nhan vien thoe ten");
    printf("\n\t4.Bang luong cua  nhan vien giam dan");
    printf("\n\t5.Xoa 1 nhan vien");
    printf("\n\t*****");
    printf("\n\tChon 1 trong chuc nang tren: ");
    scanf("%d",&ch);
    switch(ch)
    {
        case 1:
            .....
            .....
    }
}

```

2. Nhập danh sách của nhân viên.

- ✚ Đưa vòng lặp vào và gọi hàm nhập 1 nhân viên.
- 3. Xuất danh sách của nhân viên.
 - ✚ Đưa vòng lặp vào và gọi hàm xuất 1 nhân viên.
- 4. Tìm nhân viên theo tên bằng giải thuật tìm kiếm mà bạn đã được học (tìm kiếm tuần tự và tìm kiếm nhị phân).
 - ✚ Tên nhân viên cần tìm duyệt danh sách, dùng hàm **strcmp** trong thư viện **string.h** để kiểm tra 2 tên có trùng nhau không? Nếu bằng **0** thì tìm thấy, return lại vị trí tìm được, ngược lại tìm không thấy nhân viên có tên đã nhập **-1**
- 5. Tìm nhân viên theo mã nhân viên bằng giải thuật tìm kiếm mà bạn đã được học (tìm kiếm tuần tự và tìm kiếm nhị phân).
 - ✚ Mã nhân viên cần tìm, duyệt danh sách, nếu mã nhân viên bằng mã số cần tìm thì return lại vị trí tìm được, dùng lệnh break để thoát khỏi vòng lặp, ngược lại return **-1**.
- 6. Sắp xếp danh sách nhân viên theo lương bằng giải thuật sắp xếp mà bạn đã được học (Select Sort, Bubble Sort....).
 - ✚ Duyệt từ vị trí $i=0$ đến vị trí $n-1$, duyệt từ vị trí $j=i+1$ đến n . Nếu lương của vị trí $i <$ lương vị trí j thì.

```
float t = ds.a[i].luong;  
ds.a[i].luong = ds.a[j].luong;  
ds.a[j].luong = t;
```
- 7. Xóa nhân viên theo mã nhân viên.
 - ✚ Mã số nhân viên cần xóa
 - ✚ Gọi lại hàm tìm mã số nhân viên
 - ✚ Nếu hàm tìm mã số nhân viên = **-1** thì không tìm thấy nhân viên cần xóa. Ngược lại thì duyệt danh sách từ vị trí tìm được mã số đến hết danh sách **$i=i+1$ khi đó $n-1$**

Y3 – Yêu cầu về viết báo cáo

1. Sinh viên viết báo cáo toàn bộ quá trình thực hiện đề tài của mình

MÔ TẢ SẢN PHẨM PHẢI NỘP

N1 – Mã nguồn hệ thống

Folder có tên **<tên project>** chứa project hệ thống

N2 – Báo cáo

File **BaoCao.doc** chứa nội dung của **Y2**

Đóng gói tất cả các sản phẩm trên thành một file nén theo định dạng

<MaMH>_<TenMH>_<Mã SV>_ Assignment.zip để đưa lên hệ thống LMS theo yêu cầu của giảng viên. Lưu ý về cách đặt tên:

Ví dụ về cách đặt tên gói:

CTDLGT_TEOLVPT03938_Assignment.zip

THANG ĐÁNH GIÁ

A: 70%-100%	<ul style="list-style-type: none"> - Hệ thống hoạt động tốt theo các yêu cầu - Đáp ứng Y2, Y1
B: 60%-69%	<ul style="list-style-type: none"> - Đáp ứng Y2.1, Y2.2, Y2.3, Y2.4, Y2.5

C: 50%-59%	- Đáp ứng Y2.1, Y2.2, Y2.3, Y2.4
D: 40%-49%	- Đáp ứng Y2.1, Y2.2, Y2.3
D: < 40%	<ul style="list-style-type: none"> - Không đóng gói được theo yêu cầu - Hệ thống không hoạt động được - Thiếu một trong các sản phẩm: N1, N2