**GIẢI THÍCH CODE ĐỒ ÁN “THI TRẮC NGHIỆM ”**

giải thích cấu trúc dữ liệu:

-dòng 399 hàm void NhapLop():

+kiểm tra liệu số lượng lớp đã đầy chưa, nếu đầy rồi thì ngưng lại.

+ngược lại, cho nhập vào:

+nếu chuỗi rỗng thì hỏi người dùng có muốn thoát hay không

+khác rỗng: tìm mã lớp xem có bị trùng hay không, nếu trùng quay lại chỗ nhập mã lớp, đúng cho tiếp tục nhập tên lớp(kiểm tra tên lớp có bị rỗng và bị trùng không, nếu có cho nhập lại đến khi không bị lỗi)

nhập thành công thì cho FirstSV là NULL, tăng số lượng hiện tại lớp lên 1.

dòng 475 int LuuFileDSLOP(DsLop &dslop): Luu lop(ten, ma) va danh sach sinh vien trong lop

dòng 501 int XuatLop(DsLop dslop): in danh sách tất cả các lớp.

dòng 583: int NhapSV(DsLop &dsLop)

+kiểm tra mã lớp có rỗng không

+kiểm tra mã lớp có tồn tại không

nhập thông tin sinh viên(chi tiết, check lỗi)

dòng 826: int XuatDSSV\_OF\_LOP(DsLop dslop)

cho người dùng nhập vào malop, dùng vòng lặp do-while nếu:

+malop rỗng hỏi người dùng có muồn thoát hay không

+malop tìm không thấy thì cho người dùng tiếp tục nhập lại mã lớp

+nếu tìm thấy thì cho in ra danh sách sinh viên trong lớp.

=> in mã lớp xong, nếu người dùng nhấn Enter thì cho thoát.

dòng 896: void Insert\_LastDiem(Listdiemthi &First, Diemthi dt)

thêm diemthi vào cuối danh sách

//MON HOC

dòng 914: bool isEmpty(LISTMONHOC plist)

kiểm tra danh sách môn học có bị rỗng không

dòng 919: bool isFull(LISTMONHOC plist)

kiểm danh danh sách môn học có đầy không(MAX=100)

dòng 924: void create(LISTMONHOC &plist)

khởi tạo số phần tử của list là 0

dòng 930: bool kiemTraMAMH(string mamh, LISTMONHOC plist, int j)

hàm kiểm tra mã môn học có bị trùng trong danh sách hay không, với j là số phần tử đã dùng, nếu "true" là không trùng, còn false là trùng.

dòng 942: int kiemTraMAMH(string mamh, LISTMONHOC plist)

kiểm tra mã môn học đã tồn tại hay chưa, nếu rồi thì trả về vị trí, nếu chưa thì trả về -1

dòng 955: int nhapMONHOC(MONHOC &mh)

dùng 2 vòng lặp do-while, nếu mamh trống, cho nhập lại, nếu tenmh trống cho nhập lại

dòng 1006: void xuatMONHOC(MONHOC mh)

xuất tên mh và mã mh

dòng 1013: string themMONHOC(MONHOC temp, LISTMONHOC &plist)

hàm truyền vào monhoc từ trước, kiểm tra:

+danh sách đầy thì không cho thêm nữa

+kiểm tra mã môn học có bị trùng không

=>nếu đảm bảo điều kiện thì ta thêm vào môn học mới vào danh sách.

dòng 1119: void xuatLISTMONHOC(LISTMONHOC plist)

nếu danh sách trống: xuất thông báo "không có môn học nào cả"

ngược lại: cho danh sách chạy từ 0->n để in ra mamh+tenmh

dòng 1216: int Sua\_MonHoc(LISTMONHOC &dsmh)

nhập vào mã môn học, kiểm tra có tồn tại hay không:

+ nếu tồn tại cho nhập vào mã môn học mới

+kiểm tra mã môn học mới có bị rỗng hay hay không, kiểm tra mã môn học mới có bị trùng hay không, nếu không:

+cho nhập tên môn học, kiểm tra tên môn học có bị rỗng hay không, nếu rỗng thì nhập lại, nếu không rỗng thì ghi nhận môn học mới.

dòng 1320: int XoaMonHoc(LISTMONHOC &dsmh)

nhập vào mã môn học cần xóa: kiểm tra chuỗi có rỗng hay không, kiểm tra mã môn học có tồn tại hay không:

+nếu tồn tại, hỏi ng dùng có muốn xóa hay không, nếu muốn xóa thì ta xóa, còn không thì không xóa.

+nếu không tồn tại, cho người dùng nhập lại.

//CAU HOI

dòng 1406: void nhapCAUHOI(CAUHOI &ch, LISTMONHOC lmh)

+dùng vòng lặp do-while: nhập nội dung câu hỏi,lặp lại nếu nội dung câu hỏi bị bỏ trống

+nhập câu trả lời A, lặp lại nếu câu trả lời câu A bị bỏ trống.

+nhập câu trả lời B, lặp lại nếu câu trả lời câu B bị bỏ trống hoặc trùng với câu A

+nhập câu trả lời C, lặp lại nếu câu trả lời câu C bị bỏ trống hoặc trùng với câu A hoặc B

+nhập câu trả lời D, lặp lại nếu câu trả lời câu D bị bỏ trống hoặc trùng với câu A hoặc B hoặc D

sau đó cho người dùng nhập vào đáp án đúng của câu hỏi, nếu ng dùng nhập vào khác A,B,C,D,a,b,c,d thì bắt người dùng nhập lại.

dòng 1515: void xuatCAUHOI(CAUHOI ch)

xuất nội dung của một câu hỏi.

dòng 1527: void Init(TREECH & root)

hàm khởi tạo cây

dòng 1533: bool isEmpty(TREECH root)

kiểm tra cây có bị rỗng hay không

từ dòng 1541 - dòng 1653: không cần vì đây không phải là cây cân bằng.

dòng 1670: int Insert(TREECH &T, int X, CAUHOI ch)

thêm node vào cây, nhưng phải chỉnh lại vì đây không phải là cây cân bằng nữa!

dòng 1788: void lengthCAUHOIMAMH(TREECH p, int &i, string mamh)

hàm trả về số câu hỏi theo mã môn học.

dòng 1752: void xuatLISTCAUHOI(TREECH p, string mamh)

xuất các câu hỏi có mã môn học trùng với mã môn học truyền vào.

dòng 1765: void xuatLISTCAUHOI\_OF\_MONHOC(TREECH p , LISTMONHOC lmh)

xuất danh sách các câu hỏi theo môn học(là hàm nâng cấp của dòng 1752), ở hàm này có thêm kiểm tra xem mã môn học có tồn tại hay không:

+nếu mã môn học không tồn tại, cho người dùng nhập lại tối đa 4 lần.

+nếu có tồn tại, thì kiểm tra xem môn học này có câu hỏi nào hay không, nếu không thì thông báo người dùng, còn nếu có thì xuất ra.

dòng 1809: void kiemTraCAUHOI(TREECH p, CAUHOI ch, int &i)

+kiểm tra liệu câu hỏi vừa nhập vào có trùng với câu hỏi trước đó không, trùng khi nội dung câu hỏi và nội dung các câu trả lời đều giống nhau hoàn toàn, nếu trùng thì giá trị của biến i truyền vào sẽ tăng lên 1.

dòng 1823: nếu x trùng với id của node thì trả về node đó, còn nếu không thì trả về NULL.

dòng 1841: int RANDOM(), hàm này nên thay bằng hàm mà thầy cho làm bài tập về nhà.

để dùng được hàm này, ta phải có sẵn một file "so\_ngau\_nhien.txt", có 1 tham số là n và dãy chứa các số nguyên, với n là tổng số phần tử.Tạo một mảng A[] và đọc hết các phần tử của dãy, ta gán m là phần tử cuối cùng của mảng, sau đó n--, xuất lại mảng A ra file(mảng A lúc này mất đi phần tử cuối cùng so với dãy lúc ban đầu).

/////////////////////DE THI////////////////

dòng 2025: bool isEmpty(DETHI plist) kiểm tra số lượng câu hỏi có bị rỗng hay không

dòng 2030: bool isFull(DETHI plist) kiểm các các số lượng câu hỏi có bị đầy hay không

dòng 2035: void create(DETHI & plist): tạo ra đề thi mới bằng cách cho số lượng câu hỏi bằng 0.

dòng 2042: bool kiemTraDETHI(int id, DETHI plist)

kiểm tra câu hỏi trong đề thi liệu có bị trùng lặp hay không bằng cách dùng mảng nodes để chứa id của những câu hỏi, nếu bị trùng lặp thì trả về false, còn nếu không trùng thì trả về true.

dòng 2055: int themCAUHOI\_DETHI(int id, DETHI &plist)

thêm một câu hỏi mới vào đề thi, bằng cách kiểm tra liệu danh sách câu hỏi có bị đầy chưa(nếu đầy return -1), kiểm tra câu hỏi đó có tồn tại trong đề thi chưa(nếu có return -1), nếu hợp lệ thì ta thêm câu hỏi đó vào danh sách và tăng số lượng câu hỏi(biến n) lên 1 đơn vị

dòng 2070: void LayCauHoiCuaMon(int &n, int A[], TREECH lch, string mamh)

lấy câu hỏi theo môn bằng cách duyệt cây, nếu như node hiện tại đang xét vào có mã môn học trùng với mamh thì ta nhận id của câu hỏi đó vào mảng A[] và tăng giá trị n lên 1 đơn vị, tiếp tục duyệt cây sang node trái và phải.

dòng 2084: đổi vị trí của 2 số nguyên

dòng 2091: dùng để xáo trộn vị trí của mảng, mục đích nhằm lấy ra được những câu hỏi ngẫu nhiên.

Dòng 2107:dùng hàm **LayCauHoiCuaMon(n, A, lch, mamh);** để lấy được tát cả những câu hỏi của môn học; **XaoTronMang(A, n);** để xáo trộn các câu hỏi,dùng vòng lặp với số lần lặp bằng số câu hỏi để **mảng ldt** sẽ nhận số câu hỏi ngẫu nhiên tương ứng với số câu hỏi do người dùng nhập vào.

Dòng 2414: **void hienThiCAUTRALOI(int soch, string dapan**) hiển thị câu trả lời tương ứng với đáp án.

Dòng 2429: **string dapAnDETHI(DETHI dt, TREECH lch)**

tạo một chuỗi là những đáp án tương ứng với đề thi.

Dòng 2441: **bool KiemTra(Sv sinhvien, string mamh)**

Nhập vào mã mh và sinh viên, kiểm tra xem sinh viên đã từng thi môn học này chưa.

Dòng 2455: **Listdiemthi searchDIEMTHIbyMASV\_byMAMH (Listdiemthi ldt, string mamh)** truyền vào một danh sách các môn học, ta cho vòng lặp chay để kiếm tra và nhận về node điểm thi có mã mh bằng mamh ta truyền vào.

Dòng 2484: void thiTRACNGHIEM(TREECH lch, LISTMONHOC lmh, LISTSV sv , Sv \*sinhvien)

+Kiểm tra mamh liệu có tồn tại hay không, nếu không tồn tại mời nhập lại.

+cho thí sinh nhập vào số câu hỏi thi, nếu số câu hỏi<0 hay số câu hỏi>i(số câu hỏi hiện có) thì mời thí sinh nhập lại.

+cho thí sinh nhập vào thời gian.

Dòng 2540: kiểm tra danh sách sinh viên == NULL đúng không, nếu đúng thì cho Listdiemthi p ==NULL, hoặc nếu danh sách sinh viên != NULL, thì ta dùng hàm searchDIEMTHIbyMASV\_byMAMH(sinhvien->FirstDiem, mamh); để kiểm tra sinh viên đã thi môn học này chưa.

Dòng 2548:if(p==NULL) tức sinh viên chưa thi hoặc giáo viên đang thi

+ta tạo một chuỗi đáp án rỗng

+thời gian thi chuyển sang giây.

+lấy thời gian bắt đầu thi.

+lấy nội dung từng câu hỏi theo id truyền vào.

Dòng 2762: void XuatDSDiem\_Lop(DsLop dslop, LISTMONHOC plist)

Hàm xuất danh sách điểm thi theo lớp và theo môn học.

+nhập vào mã lớp, kiểm tra lớp học có tồn tại.

+nhập vào mã môn học, kiểm tra mã môn học có tồn tại.

Nếu tồn tại lớp và mã môn học do người dùng nhập vào, ta lấy danh sách sinh viên thuộc lớp đó và duyệt-in ra chi tiết thông tin sinh viên và môn thi.

Dòng 2857: void xuLiMenuGV(DsLop &dslop, LISTMONHOC &lmh, TREECH &lch)

Thiết kế giao diện khi đăng nhập bằng tài khoản giảng viên.

Dòng 2971: void design\_DangNhap()-thiết kế giao diện đăng nhập đầu tiên.

Dòng 3008: LISTSV xuLiDangNhap(DsLop dsLop, int &i, Sv\* &sv)

Cho người dùng nhập vào username và password.

+if (username == "GV" && password == "GV") thì cho i=1 và return NULL.

+còn nếu không thỏa điều kiện if bên trên thì cho duyệt từng lớp, trong mỗi lớp duyệt từng sinh viên để kiểm tra username và password. Nếu tìm thấy cho i=0, sv=&(vị trí của sinh viên đó) và trả về danh sách của lớp đó. Ngược lại sau khi đã duyệt hết mà vẫn không tìm thấy ta gán i=-1, sv=NULL, và trả về NULL.

Dòng 3057: void GiaiPhongCay(TREECH &root) xóa hết các node trong cây.

Dòng 3074: void GiaiPhong(LISTMONHOC &lmh, DsLop &dslop, TREECH &cauhoi) hàm giải phóng bộ nhớ, bao gồm: giải phóng danh sách các môn học; giải phóng danh sách sinh viên từng lớp và điểm các sinh viên của các sinh viên;giải phóng cây câu hỏi.

Dòng 3099: hàm main()

+Đọc danh sách môn học, danh sách lớp, danh sách câu hỏi từ file.

+Hiển thị menu đăng nhập **xuLiDangNhap(dslop, i, sinhvien);** nếu:

+p==NULL và i==1 thì chuyển sang giao diện của giảng viên.

+i==0 và sinhvien ==1 thì cho thi trắc nghiệm ngay lập tức, sau khi thi xong thì lưu file dsLop.

+các trường hợp còn lại là đăng nhập sai, nên ta thông báo cho người dùng.

Sau khi chương trình chạy xong thì ta giải phòng vùng nhớ.

CÁC HÀM LƯU FILE:

int LuuFileDSLOP(DsLop &dsLop);

void DocFileDSLOP(DsLop& dslop);

//ghi file listmonhoc ra Dsmonhoc.txt

int writefileLISTMONHOC(LISTMONHOC plist)

//đọc file listmonhoc từ Dsmonhoc.txt

int readfileLISTMONHOC(LISTMONHOC &plist)

//ghi câu hỏi ra file

void writefileLISTCAUHOI(TREECH p, ofstream &fo)

int writefileLISTCAUHOI(TREECH p)//hàm này kiểm tra file trước khi ghi

int readfileLISTCAUHOI(TREECH &p)// đọc danh sách câu hỏi từ file filelistcauhoi.txt

lưu ý: kiểm tra hàm trùng dòng 930><dòng 942

kiểm tra dòng 1304: liệu có viết thừa.

dòng 2070: void LayCauHoiCuaMon(int &n, int A[], TREECH lch, string mamh) chưa hiểu lắm tác dụng của biến n

dòng 2091: cần giảng rõ lại cho Tuyền

dòng 2484: **void thiTRACNGHIEM(TREECH lch, LISTMONHOC lmh, LISTSV sv , Sv \*sinhvien)** cần giải thích lại cho Tuyền nắm rõ.