//chỉ up function/operator đại diện nếu có nhiều overload

STAFF FUNCION

void AddClass(\_Class\*& cls, Classes sourse)

Classes MakeClass(\_Student\*& all, bool cls, int x, int y)

void SaveClass(Classes cl, const char\* fileout)

void PrintClass(Classes a)

SchoolYear\* AddSchoolYear(int year)

void SaveSchoolYear(SchoolYear\* sch)

void AddCourse(\_Course\*& courselist, Course\* course)

Course\* MakeCourse()

STUDENT FUNCTION

Student\* StringToStudent(std::wstring str)

void AddStudent(\_Student\*& studentlist, Student\* student)

\_Student\* FileInStudent(std::string filename)

void FileOutStudent(\_Student\* stu, std::string fileout)

void StuToBin(Student\* stu, std::string fileout)

Student\* BinToStu(std::string filein)

void SaveNewStu(\_Student\* stu, char\* direction)

int NumberOfStudent(\_Student\* stu)

void PrintStu(Student\* a)

Student\* FindStudent(\_Student\* studentlist, unsigned \_\_int64 ID)

void DealloStu(\_Student\*& stu)

void ConnectStudent(\_Student\*& a, \_Student\*& b)

int CountStudent(\_Student\* a)

\_Student\* TypeInStudent()

void ChangeInfo(Student\*& CurrentUser);

void SaveCourseRegHis(char\* AccountUsername, Course course)

void SaveCourseCancelHis(char\* AccountUsername, Course course)

void LoginStu(Student\*& CurrentUser)

void Logout(Student\* CurrentUser)

void registerCourse(Course\*\*newReg,std::string sem,unsigned \_\_int64 stuID) {

void cancelCourse(Course\*\* cancelReg, std::string sem, unsigned \_\_int64 stuID) {

void registerCourse(Student\*stu,Course\*\*cou,std::string sem) {

void ChangePassword(Student\* CurrentUser)

void ChangeSocialID(Student\* CurrentUser)

void ChangeDOB(Student\* CurrentUser)

void ChangeInfo(Student\*& CurrentUser)

\_Student\* SearchStuList(unsigned \_\_int64\* ID) // Return LL \_Student // Use dynamic array of ID when using

Student\* SearchStu(unsigned \_\_int64 ID)

Student\*\* SearchStuArr(unsigned \_\_int64\* ID, std::string Grade) // Use dynamic array of ID when using

DATA FUNCTION

bool AddStudentWarning(unsigned \_\_int64 ID)

std::string NumToDay(unsigned short day)

void SessionConflictWarning(Course\* a, Course\* b) // false means conflict, true is not

bool isConflict(Course\* a, Course\* b) // true means conflict, false is not

void SessionConflict(Course\*\* a, Course\*\* b, int\*& Register)

std::string GetFilePath(unsigned \_\_int64 ID)

void classifyCourse(Course\*\* reg, Course\*\* wasReg, Course\*\*& cancelReg, Course\*\*& newReg)

wchar\_t\* RemoveSpace(wchar\_t\* a)

Course\* searchID(\_Course\* courselist, std::wstring search)

Course\* searchName(\_Course\* courselist, std::wstring search)

Course\* searchTeacher(\_Course\* courselist, std::wstring search)

Course\* searchCourse(\_Course\* courselist, std::wstring& search)

void printCourseDay(short a)

void displaylistCourse(Course\* cou)

void displayCourse(Course\* cou)

void displayScore(Score a)

void editCourse(Course\* cou, std::string filename, std::string current)

Date TakeDateEnd(std::string current)

Date TakeDateStart(std::string current)

void SaveLoginHistory(char\* AccountUsername)

void pickSchedule(Course\* cou, int x, int y)

void editInfo(Course\* cou, std::string filename, std::string current)

void editScore(Course\* cou, std::string filename, std::string current)

void FileInScore(Course\* cou, std::string direction)

float StringToFloat(wchar\_t\* ch)

void StringToScore(Score& a, std::wstring str)

void editCourse(std::string filename,std::string current)

void deleteCourse(Course\* cour, std::string filename, std::string current)

Course\* searchCourseFile(std::string search )

//FILE

Filelist\* TakeFileInFolder(const std::wstring& name)

int countFile(Filelist\* list)

void DeleteCurFileList(Filelist\*& filelist)

void GotoXY(short x, short y)

//OPERATOR OVERLOAD

std::ostream& operator<<(std::ostream& os, const Date& dt)

bool operator==(Date a, Date b)

bool operator<=(Date a, Date b)

bool operator>=(Date a, Date b)

std::string ViewSemesterTime()

void AddInListFile(Filelist\*& direc, std::string add)

std::string TakeCurrent()

std::string ChoseFolder(Filelist\* list, short x, short y)

std::string InputHidden()

void SaveScore(Course\* cou, std::string filename)

void CourseToBin(Course\* course, std::string filename, std::string current)

Course\* BinToCourse(std::string filename)

INTERFACE FUNCTION

//menu

short Menu(wchar\_t\*\* list, short Xposition, short Yposition)

short Choose(wchar\_t\*\* list,short X,short Y)

bool isLoggedIn();

void enrollCourse(Student student, Course course);

void updateStudentsInClass(Classes& a);

void userTypeMode();

void studentMode(Student\* stu=nullptr);

void staffMode();

void studentMenu();

void gradeMenu();

void classMenu();

void addGrade();

void staffStudentMenu();

void ViewGrade();

void AddClass();

void ViewClass();

void StuInformation(Student\* stu);

void CourseInformaion(Student\* stu);

void RegisterCouse(Student\* stu);

void ViewCouse(Student\* stu);

void courseStaff();

void schoolPlan();

void addCourse();

void editCourse();

void viewCourse();

void removeCourse();

void addStuInClass();

void viewStu();

void OpenRegister();

SUPPORT FUNCTIONS

unsigned short LStringToPerform(wchar\_t\* temp)

Course\* StringToCourse(std::wstring str)

void FileOutCourse(\_Course\* cou, std::string fileout)

//unicode

void \_LText()

void \_SText()

//str

void LStrToStr(char desination[], int size, std::wstring source)

void StrToLStr(wchar\_t desination[], int size, char\* source)//đại diện overload mấy cái còn lại

void StrCat())//đại diện overload mấy cái còn lại

char\* AddTwoStr(const char\* a, const char\* b)

char\* StrToChar(std::string source)

std::string ToString(std::wstring source)

std::wstring ToWstring(wchar\_t\* source)

//

unsigned \_\_int64 StringToInt

unsigned \_\_int64 InputNumber()

Date StringToDate(const wchar\_t\* ch)

Date InputDate()

bool LeapYear(int y)

bool CheckDate(Date dat)

bool operator>(tm& t1, tm& t2) //compare date

std::string NumToStr(unsigned \_\_int64 num)

char\* AddTwoStr(const char\* a, const char\* b)

void MakeCurentTime(int year)

int CountFile(Filelist\* a)

SYSTEM FUNCTION

void removeAcourseForStudents(int studentsIDs[],char\* courseID)

OPTIMIZE FUNCTION

void DealocatedArrString(char\*\*& stringarr)

void deleteStu(Student\*& a)

void delete\_Stu(\_Student\*& a)

void deleteCourse(Course\*& a)

void delete\_Course(\_Course\*& a)

void deleteClasses(Classes\*& a)

void delete\_Class(\_Class\*& a)

void deleteSemester(Semester\* a)

void deleteSchoolyear(SchoolYear\*& a)

void deleteFilelist(Filelist\*& a)

void deleteIntArray(int\*& a)

void deleteStuArray(Student\*\*& a)