

Thi giữa kỳ môn KTLT lớp 22CLCA

1 Thông tin đề thi:

- Thời gian làm bài : **75 phút**
- Làm bài và nộp theo deadline trên Moodle

2 Yêu cầu:

- Đặt tên Project là MSSV (VD : 22127001)
- Tạo file cpp đặt tên là MSSV.cpp (Ví dụ : 22127001.cpp). ***Có thể tạo những file khác tùy ý.***
- Đầu file MSSV.cpp ghi rõ thông tin như sau:

```
/*  
Ho va ten :  
MSSV :  
*/
```

```
Ví dụ  
/*  
Ho va ten : Nguyen Van A  
MSSV : 22127001  
*/
```

- Ghi chú :

- Chỉ nộp bài bao gồm các file: **.cpp, .h, .vcproj và .sln**.
- Lưu bài trong quá trình làm (Nhấn Ctrl + S).
- **Xóa bài sau khi nộp.**
- **Có dấu hiệu gian lận sẽ 0 điểm, không chấp nhận bất kỳ lý do gì.**
- **Chương trình build lỗi sẽ 0 điểm.**
- KHÔNG được phép sử dụng các hàm tính toán cơ bản (min, max, pow,...)
- Chỉ sử dụng trong số các thư viện sau: <iostream>, <fstream>, <sstream>, <stdio.h>, <string.h>
- Cho phép sử dụng duy nhất website: www.cppreference.com. Nghiêm cấm sử dụng mạng xã hội, tutorial.

3 Đề bài

Viết chương trình với các yêu cầu sau:

- Cấu trúc kiểu dữ liệu **HocSinh** như sau:

```
struct HocSinh
{
    char *hoTen;
    char *maSo;
    double diemTB;
};
```

- Cấu trúc kiểu dữ liệu **Option** như sau:

```
struct Option
{
    int min, max;
};
```

- [1 điểm]: Viết hàm tạo giá trị ngẫu nhiên làm tròn đến .125 (tức là mỗi lần gọi hàm sẽ được 1 giá trị **double** khác nhau với phần thập phân là: .0 / .125 / .25/... / .875) trong khoảng cho trước [**min**, **max**] với khai báo hàm như sau:

```
double GenerateRandom(const Option &opt);
```

Gợi ý: sử dụng hàm **rand()** để lấy giá trị **int** ngẫu nhiên và ràng buộc giá trị (bao gồm chuyển số thực, làm tròn) đó lại trong khoảng **min**, **max**.

- [1 điểm]: Viết hàm rút trích dữ liệu từ 1 chuỗi biết rằng dữ liệu đầu vào là **char*** có định dạng "**<họ tên>-<mã số>**" và cần lấy thông tin là chuỗi Họ tên và chuỗi Mã số (đều là **char***). Với yêu cầu khai báo hàm như sau:

```
void ExtractInfo(const char *strInput, char* &pHT, char* &pMS);
```

Ví dụ: chuỗi đầu vào là "Nguyễn Van A-22127001" thì cần lấy được pHT là "Nguyễn Van A" và pMS là "22127001".

- [1 điểm]: Viết hàm tạo **HocSinh** từ chuỗi **char*** dữ liệu đầu vào định dạng như câu 2 và phát sinh điểm trung bình cho học sinh đó trong khoảng yêu cầu [**dtbMin**, **dtbMax**] với khai báo hàm như sau:

```
HocSinh* CreateHocSinh(const char *strInput, const double &dtbMin,
const double &dtbMax);
```

- [1 điểm]: Viết hàm giải phóng **HocSinh** (thu hồi hoàn toàn bộ nhớ) với khai báo hàm như sau:

```
void FreeHocSinh(HocSinh *pHS);
```

5. [1 điểm]: Viết hàm xuất **HocSinh** ra chuỗi **char*** với định dạng:

"<họ tên> - <mã số> [<điểm trung bình>]"

với khai báo hàm như sau:

```
char* HocSinhToString(const HocSinh *pHS);
```

Ví dụ: "Nguyen Van A - 22127001 [8.375]"

6. [1 điểm]: Viết hàm đọc danh sách **HocSinh** từ file **"data.txt"** với khai báo hàm như sau:

```
void Load(const char *path, HocSinh** &data, int &nCol, int &nRow)
```

Biết các **Học sinh** được xếp vào vị trí như ma trận và cấu trúc file như sau: **dòng đầu** là giá trị **int** của số hàng (nRow), **dòng** tiếp theo là giá trị **int** của số cột (nCol), các **dòng** còn lại là thông tin của **HocSinh** theo định dạng ở câu 2.

7. [1 điểm]: Viết hàm giải phóng toàn bộ các **HocSinh** (thu hồi hoàn toàn bộ nhớ) với khai báo hàm như sau:

```
void FreeData(HocSinh** &data, const int &nCol, const int &nRow)
```

8. [1 điểm]: Viết hàm xuất danh sách **HocSinh** ra màn hình theo đúng vị trí của **HocSinh** và chia tách bởi **TAB**, định dạng mỗi **HocSinh** như câu 5 với khai báo hàm như sau:

```
void PrintData(HocSinh** data, const int &nCol, const int &nRow)
```

Ví dụ: *Nguyen Van A – 22127001 [9.25]* *Tran Van C - 22127002 [5.875]*
Le Thi C - 221270003 [8.125] *Pham Thi D – 221270004 [6.75]*

9. [2 điểm]: Viết hàm main như sau và bổ sung hàm còn thiếu.

```
void main()
{
    srand(7103);
    const char* path = "data.txt";
    HocSinh** data = NULL;
    int nCol = 0, nRow = 0;
    Load(path, data, nCol, nRow);
    PrintData(data, nCol, nRow);
    cout << endl << endl;

    /// xuất ra màn hình Học sinh có điểm trung bình lớn nhất
    /// cần thông tin vị trí của Học sinh
    PrintTheBestHocSinh(data, nCol, nRow);

    FreeData(data, nCol, nRow);
    system("pause");
}
```