

ANALYSIS REPORT

How does my program work?

İlk olarak csv dosyalarından okumaları yaparak verileri structure lara okuyorum.Daha sonra sıralama fonksiyonu ile öğrencilerin puanlarını sıralıyorum.(Çünkü puanı yüksek olanın yerleşme hakkı daha fazla). Öğrenciler puanı büyükten küçüğe doğru sıralıyorum ve sıralama yaparken eğer 2 sayısal öğrencisi karşılaştırılacak ise sadece sayısal puanlarını karşılaştırıp skoru büyük olanı temp yaparak yer değiştiriyorum. Eğer karşılaştırdığım 2 öğrenciden biri sayısal diğeri sözel ise sözel puanlarını karşılaştırarak büyük skorlu olanı üste alıyorum.Eğer karşılaştırdım öğrencilerin ikisi de sözel ise sözel puanlarını karşılaştırıyorum ve bu şekilde en büyük skordan en küçüğe skorlar sıralanmış oluyor.Bir de eğer öğrencilerin puanları eşitse doğum günlerini , eğer onlarda eşitse random şekilde yerleştiriyor. Daha sonra bu sıraya uyarak başka bir fonksiyon ile öğrencilerin yerleşmelerini yapıyorum. Eğer kişinin ilk tercihindeki departmanın konteymanı dolu ise 2. Tercih, ... aynen böyle devam ediyor ta ki tercihlerden biriyle yerleşene kadar ya da bütün üniversitelerin konteymanları dolmuş ise kişi hiç bir yere yerleşemiyor...Yerleşimleri yaptıktan sonra da kullanıcıdan bazı veriler alıp kullanıcıya sınav sonuçları hakkında bazı istediği verileri yazdırıyorum bu yerleştirme işlemleri devam ederken de bazı verileri binaryFile lerde güncel bir şekilde tutmaya çalışıyorum. Ve son olarak da kullanıcının sonuçları daha rahat görebileceği csv output dosyaları oluştuyorum. Bu verileri bir de html sayfalarını yaparken kullanıyorum...

Inputs/Outputs(I/O)

Input Files:

- StudentsNames.csv*:A csvfile to hold student names
- StudentsAdressInfo.csv*:A csv file to hold addressing information
- Departments.csv*:A csv file which holds all data about universities, their departments and the quotas of the departments.
- Scores.csv*:A csv file which holds the scores of students
- Preferences.csv*:A csv file that holds the preferences of the students.

Outputs:

- PlacementToDepartments.csv*:The file which holds the departments and the students placed to them.

- PlacementOfStudents.csv*: The file which holds the students and the departments they are placed.
- PlacementToAUni.csv*: The file which includes names and scores of all students placed into a specific university
- lastStatusOfDepartments.csv*: The file which includes maxscore ,minscore, quota and empty quota of the departments.
- Placement.html : This file will include all students, their scores and all placement information
- Students.html: This file will include all students and their personal information
- Some Binary Files (to save the placement and last situations)

#Constant Macros

```

MAXPREF          8

MAX_UNI_NUM      50

MAX_DEP_NUM      20

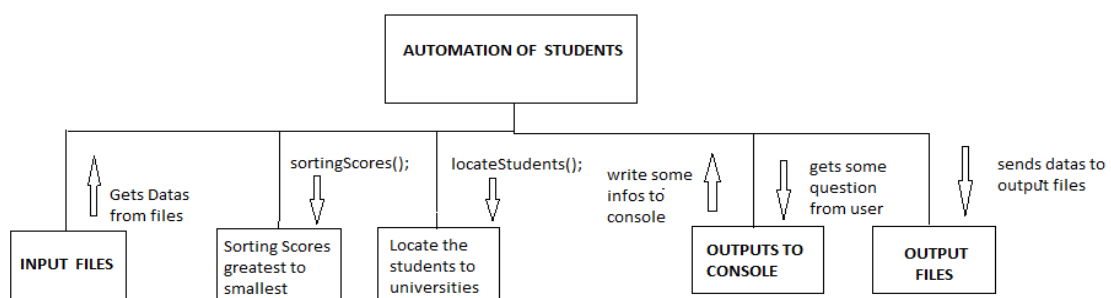
MAX_STU_NUM      100

DEBUG /* For debug mode */

EMPTY            -1

```

STRUCTURE CHART



STRUCTURES

```
#typedef struct{
```

```
    int day;
```

```
    int month;
```

```
    int year;
```

```
}birth_t;
```

```
#typedef struct{
```

```
    char uniPref;
```

```
    char depPref;
```

```
}pref_t;
```

```
#Typedef struct{
```

```
    char IDnum[11];
```

```
    char name[20];
```

```
    char sirname[20];
```

```
    birth_t birthday;
```

```
    char scrType;
```

```
    char uniName[30];
```

```
    char depName[30];
```

```
    int prefNum;
```

```
    pref_t preferences[MAXPREF];
```

```
}stuInfo_t;
```

```
#typedef struct{
```

```
    char IDnum[11];
```

```
    char phone[11];
```

```
    char mail[20];
```

```
    char address[40];
```

```
}stuAddress_t;
```

```
#typedef struct{
```

```
    char IDnum[11];
```

```
    double Qscore;
```

```
    double Vscore;
```

```
    int uniqueID;
```

```
}score_t;
```

```
#typedef struct{
```

```
    char depName[30];
```

```
    int depID;
```

```
    char scrType;
```

```
    int quota;
```

```
    int emptyQuota;
```

```
    double minScore;
```

```
    double maxScore;
```

```
}dep_t;
```

```
#Typedef struct{
```

```
    char uniName[30];
```

```
    int uniID;
```

```
    dep_t departs[MAX_DEP_NUM];
```

```
}uni_t;
```

FUNCTIONS PROTOTYPES

```
**int readstuInfos( char stuInfoFile[], stuInfo_t stuInfo[] ); // reads the StudentNames.csv and write that's content to struct array..
```

```
**void printstuInfos( stuInfo_t stuInfo[], int number ); // prints the content of structure array(stuInfo[] ) to console
```

```
**int readDepartments( char departsFile[] ,uni_t universities[MAX_UNI_NUM], int numberOfUni, int* numberOfDeps );
```

```
// reads the Departments.csv and write that's content to struct array..
```

```
**void printDepartments( uni_t universities[MAX_UNI_NUM], int number );
```

```
// prints the content of structure array(universities[]) to console
```

```
**int readPreferences( char prefsFile[], stuInfo_t stuInfo[20], int numOfStu );
```

```
// reads the Preferences.csv and write that's content to struct array..( stuInfo[].preferences )
```

```
**int readScores( char scoresFile[], score_t scores[], int numOfStu );
```

```
// reads the Scores.csv and write that's content to struct array..( scores[] )
```

```
**int readstuAddress( char stuAddressFile[], stuAddress_t stuAddress[], int numOfStu );
```

```
// reads the StudentsAddressInfo.csv and write that's content to struct array..( stuAddress[] )
```

```
**void printStuAddress( stuAddress_t stuAddress[], int numOfStu );
```

```
// prints the content of structure array(stuAddress[]) to console
```

```
**int sortingScores( score_t scores[], stuInfo_t stuInfo[], int numOfStu );
```

```
//This function sorts the scores from greatest to smallest ..
```

```
**int locateStuToUni( scores_t scores[], stuInfo_t stuInfo[], uni_t universities[], int numOfStu, int numOfuni );
```

```
// This function locates the students to universities according to their scores and preferences
```

```
//For Console Outputs...
```

```
**int consAns1( char name[], char sirname[], long long int ID, stuInfo_t stuInfo[], int numOfStu );
```

```
**void consOut1( uni_t universities[], int numOfUni );
```

```
**void consOut2( uni_t universities[], int numOfUni );
```

```
**void consOut3( uni_t universities[], int numOfUni );
```

```
**int consAns2( char uniName[], stuInfo_t stuInfo[], uni_t universities[] , int numOfUni );

**int consAns3( char depName[], stuInfo_t stuInfo[], uni_t universities[] , int numOfStu );

//For CSV outs...

** void CsvOut1( uni_t universities[], stuInfo_t stuInfo[], int size );

**void CsvOut2( stuInfo_t stuInfo[], int size );

**void CsvOut3( stuInfo_t stuInfo[], int size );

**void CsvOut4( uni_t universities[] , int size );

//For Creating html files

**int createHtmlPlacement( stuInfo_t stuInfo[], int numOfStu );

** int createHtmlStudents( stuInfo_t stuInfo[], stuAddress_t stuAddress[], int numOfStu );
```