# Counselling System

UCS2201 – Fundamentals and Practice of Software Development

A PROJECT REPORT

Submitted By

Shyam Sainarayanan Varadharajan 3122225001134

Sachin Rangabaskar 3122225001116

Department of Computer Science and Engineering

Sri Sivasubramaniya Nadar College of Engineering

(An Autonomous Institution, Affiliated to Anna University)

Kalavakkam – 603110

July 2023

**Sri Sivasubramaniya Nadar College of Engineering**

**(An Autonomous Institution, Affiliated to Anna University)**

## BONAFIDE CERTIFICATE

Certified that this project report titled "**Counselling System**" is the bonafide work of "Shyam Sainarayan Varadharajan (3122225001134) and Sachin Rangabaskar (3122225001116)" who carried out the project work in the UCS2201 – Fundamentals and Practice of Software Development during the academic year 2022-23.

Internal Examiner External Examiner

Date:

# TABLE OF CONTENTS

**Introduction** Page No

.

The Joint Seat Allocation Authority (JoSAA) is a centralized counseling system in India for admissions to undergraduate engineering and architecture programs at participating institutes. JoSAA was established to simplify and streamline the admission process, bringing together various government-funded institutions under a single platform. It aims to ensure a fair and transparent allocation of seats based on the candidates' preferences and merit.

The counseling process through JoSAA begins after the announcement of results for national-level entrance exams such as JEE Main and JEE Advanced. Candidates who qualify for these exams become eligible for JoSAA counseling. The participating institutes include the Indian Institutes of Technology (IITs), National Institutes of Technology (NITs), Indian Institutes of Information Technology (IIITs), and other government-funded technical institutions.

During the counseling process, candidates are required to register and fill in their preferred choices of courses and institutes. The choices are based on their rank and eligibility criteria. JoSAA then conducts multiple rounds of seat allocation where candidates are allotted seats in their preferred courses and institutes based on their rank, category, and availability of seats. Candidates can accept the allocated seat or choose to float, slide, or freeze their options in subsequent rounds.

The counseling process also provides opportunities for candidates to upgrade their allotted seats based on their preferences and availability. It ensures that candidates have the flexibility to choose the best available option according to their aspirations and performance.

The JoSAA counseling system has significantly simplified and streamlined the admission process for undergraduate engineering and architecture programs in India. It has brought transparency, fairness, and efficiency to the allocation of seats, providing candidates with equal opportunities to secure admissions in their desired institutions based on their merit and preference.

**Problem Statement**

Develop a software system for the engineering counselling and admission process for two sets of institutes (for example, say IITs and NITs) each having a set of different branches, each branch with a certain number of seats available. Number of candidates can be assumed as 5 times the total number of seats available. Each candidate can provide a list of preferences where each preference is a 2-tuple, (institute, branch). Admission to each set of institutes is based on its own qualifying exam (for example, JEE-Advanced and JEE-Main). Each candidate will have a specific rank in one or both merit lists.

**Exploration Beyond the problem Statement**

The Gale-Shapley algorithm, also known as the Stable Marriage algorithm, is not directly used by JoSAA (Joint Seat Allocation Authority). However, the algorithm's underlying principles of stable matching and preference-based allocation are applicable to the seat allocation process carried out by JoSAA.

JoSAA uses a similar approach to the Gale-Shapley algorithm to allocate seats to candidates based on their preferences and merit. The key steps of the algorithm, which are reflected in the JoSAA process, are as follows:

Initialization: JoSAA begins by collecting preferences from both candidates (students) and institutions (colleges). Candidates rank their preferred institutes, and institutions rank their preferred candidates based on merit and other criteria.

Proposal Phase: In the Gale-Shapley algorithm, candidates propose to their preferred institutions, and institutions tentatively accept proposals from their preferred candidates until all available seats are filled. Similarly, in JoSAA, candidates provide their preferred choices of institutes during the counseling registration process.

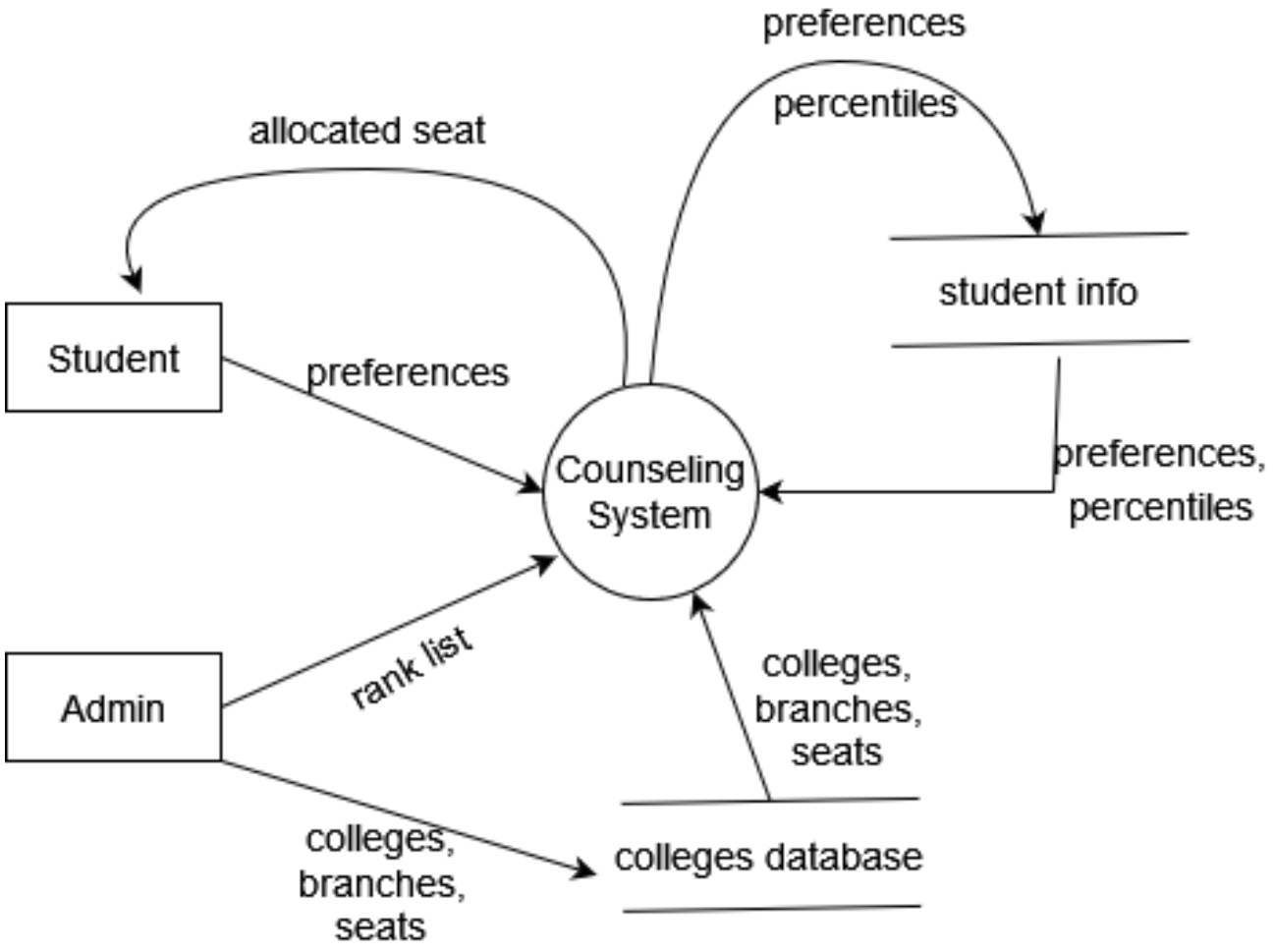
Seat Allocation: Based on the preferences and merit of candidates, JoSAA allocates seats to the candidates in order of their ranks. The allocation is carried out in multiple rounds, allowing candidates to modify their choices based on the available options. This process aims to maximize the satisfaction of both candidates and institutions.

Acceptance and Confirmation: After each round of seat allocation, candidates have the option to accept the allocated seat or choose to float, slide, or freeze their options. This flexibility allows candidates to explore better options and upgrade their seats in subsequent rounds, similar to the preference changes in the Gale-Shapley algorithm.

The primary goal of both the Gale-Shapley algorithm and JoSAA is to achieve stable matching, where no candidate and institute have an incentive to change their allocated seats.

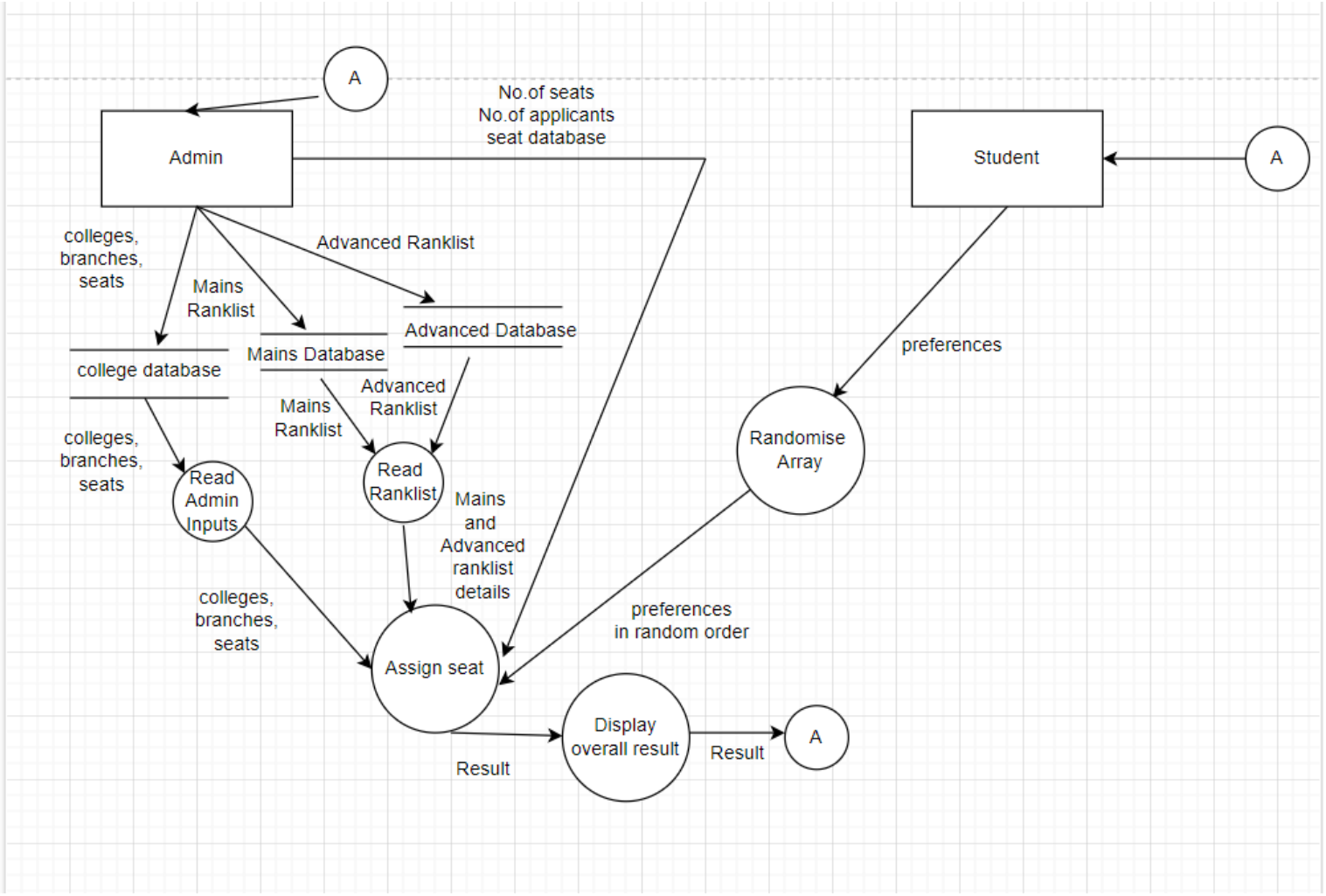
**Analysis Using Data Flow Diagrams**

**Level 0**




This above diagram underlines the overall process of this counselling system. Firstly preferences and percentiles are accepted from the student and college details are accepted from the admin. These data pass through the counselling system module and finally we get the allotted seat for the student .

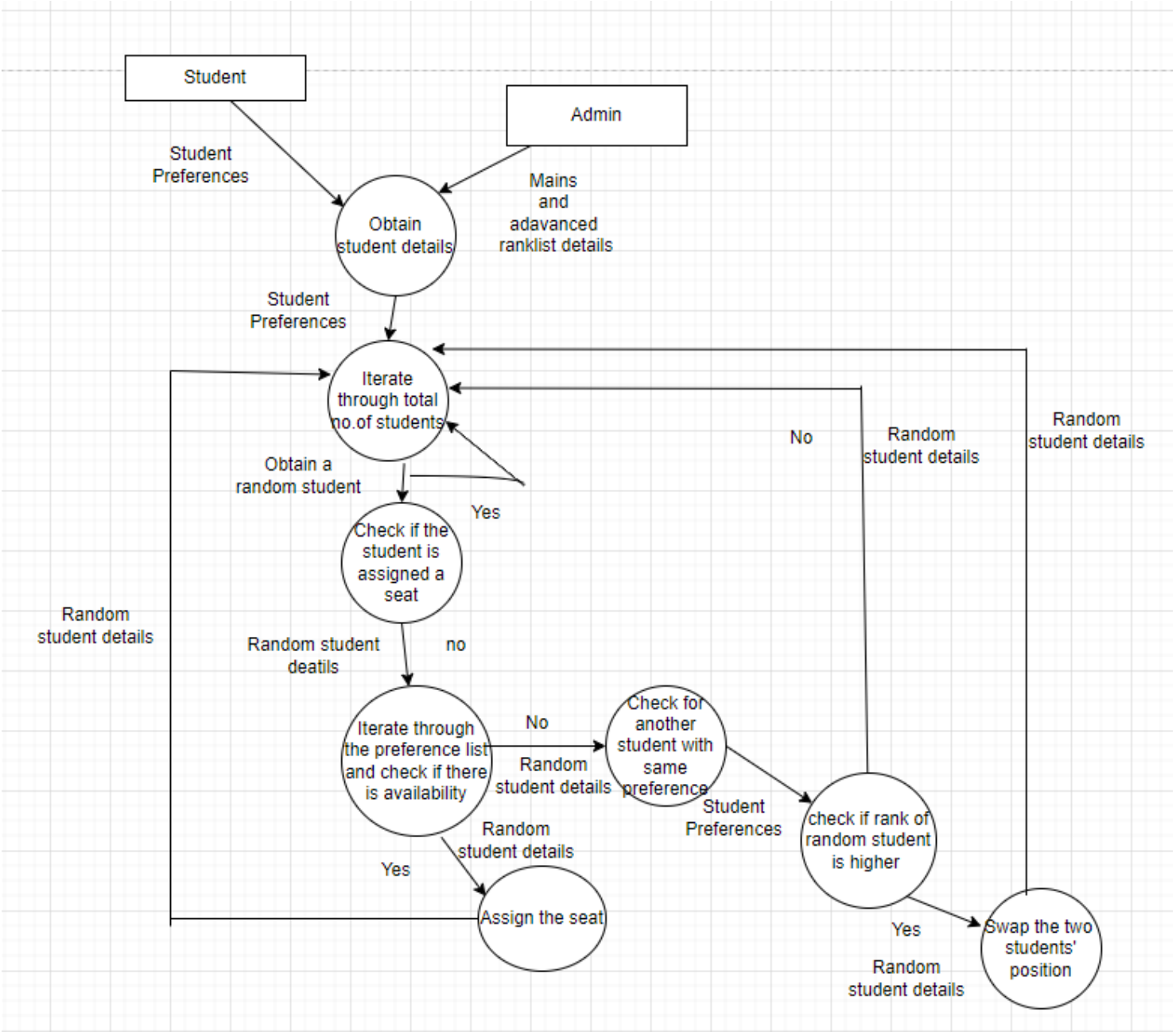
**Level 1**




Inside the counselling system module we firstly accept data from the student and admin and store it in separate data structures.

This data structure is randomised and passed to the assign seat function which allots a particular seat to the student if available.

**Level 2**

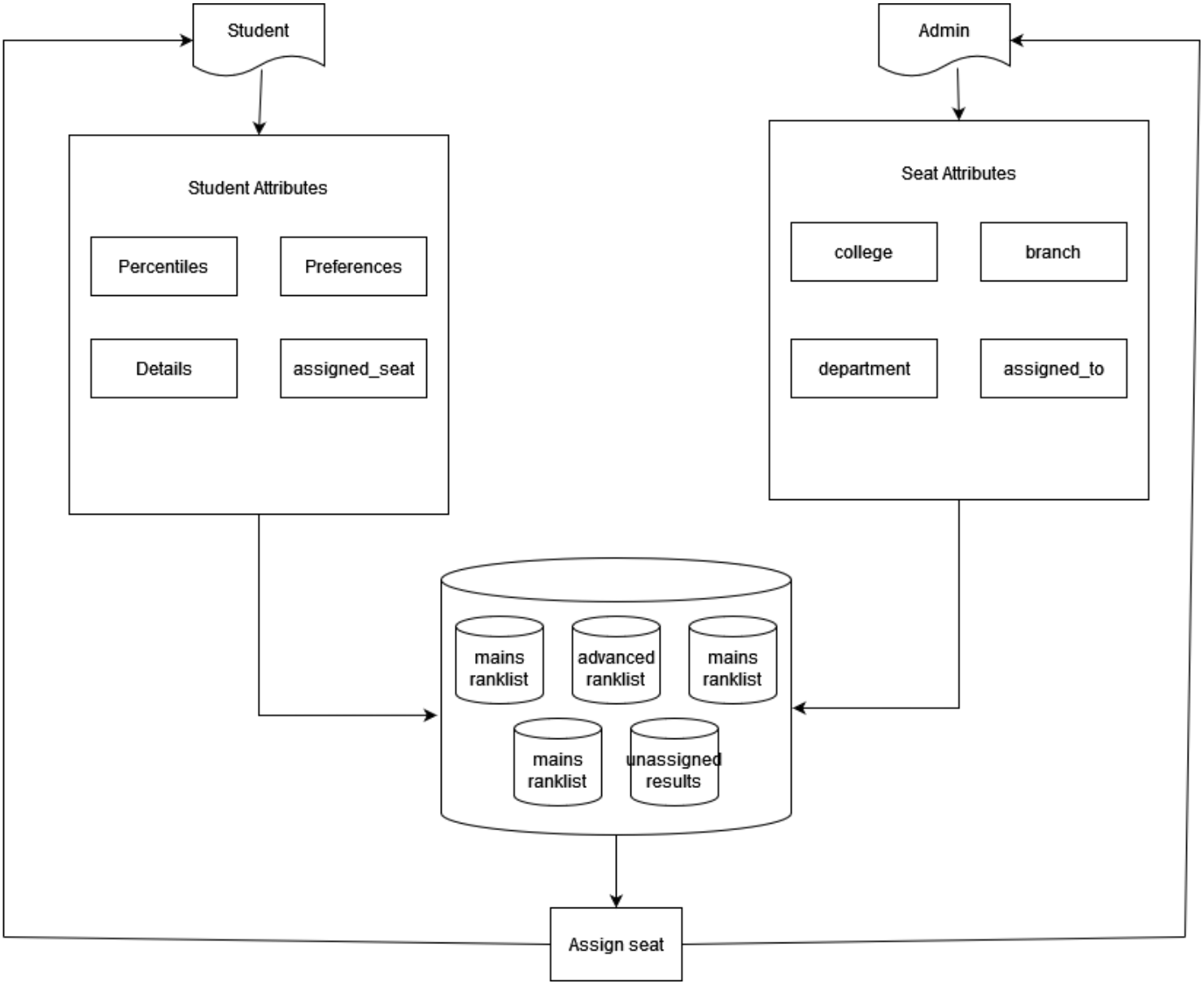



The above diagram expands the assign seat module.

Initially the randomised structure is iterated and a random student is obtained. Then the module checks if the student has a seat assigned.If yes then it skips to the next iteration or else it checks for the availability of his preferences.If his preference is available then it assigns him the seat or else checks whether there is some other person with the same preference and a lower rank who has his seat allotted.If yes then it swaps the two students or else it skips to the next iteration.

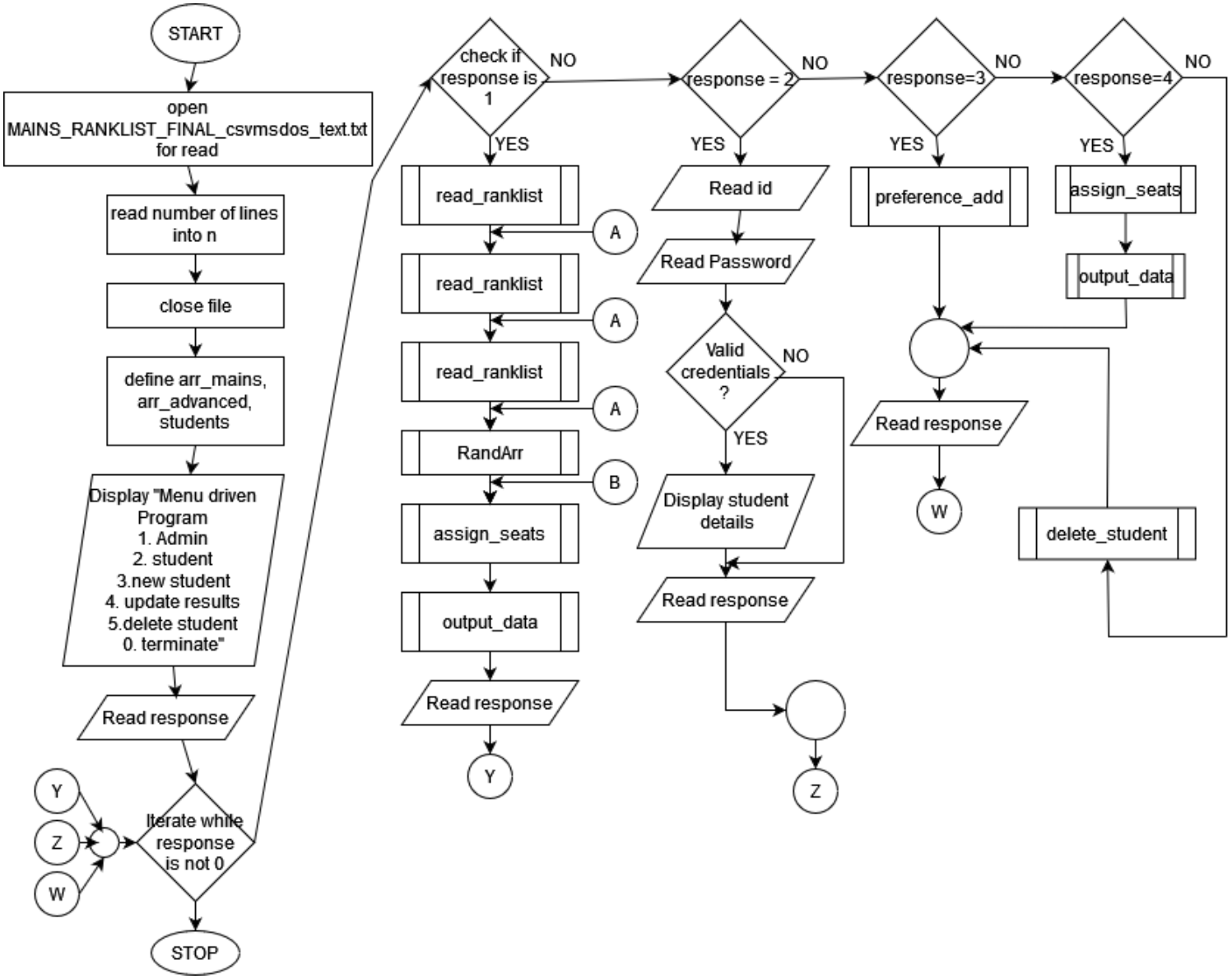
## Detailed Design

## System Architectural Diagram



## Flow Charts for modules

### Overall Flowchart

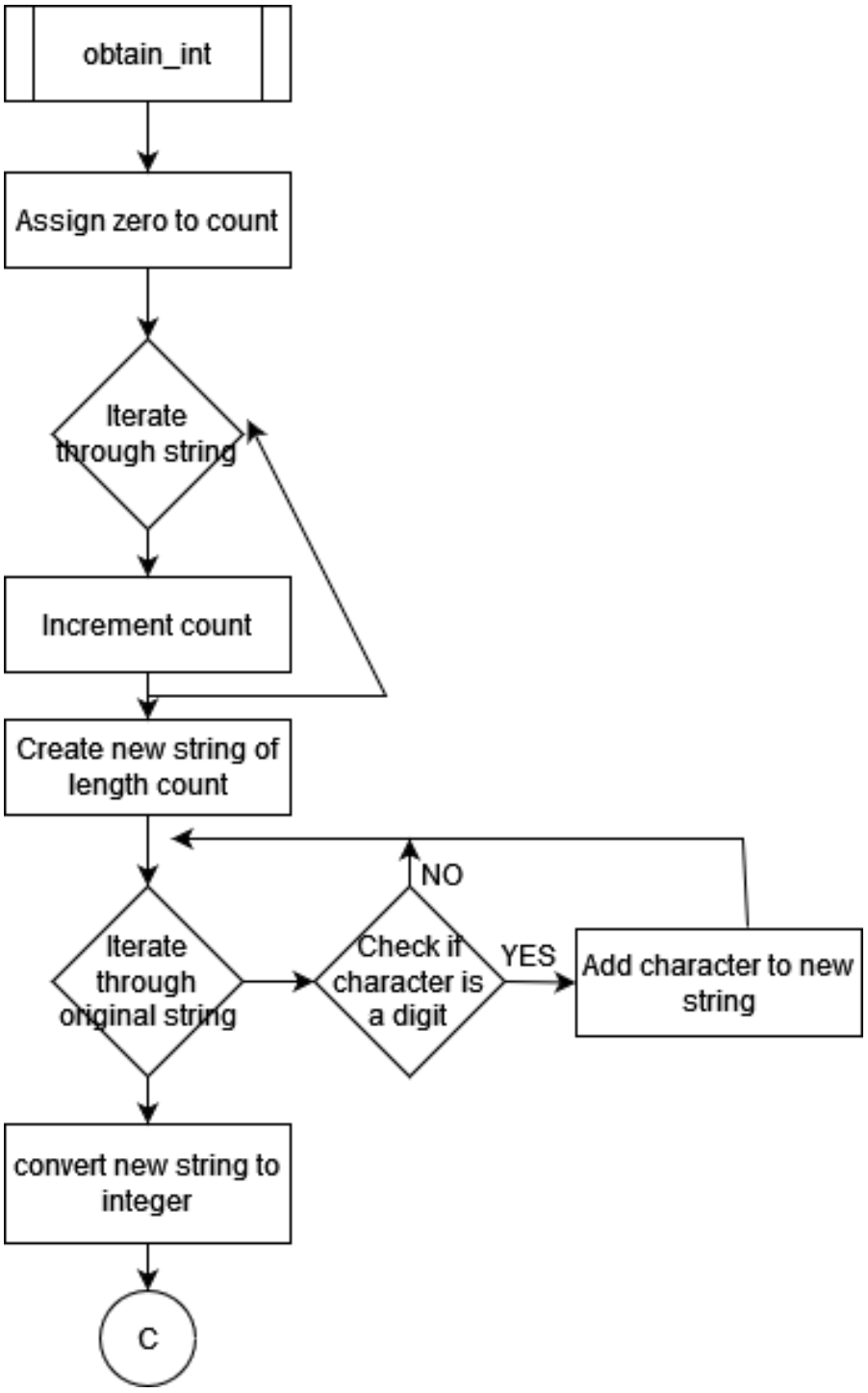



This Flowchart represents how the overall data gets converted into the required output after various stages of data processing.

It consists of 8 different modules.

These modules are expanded in the further pages.

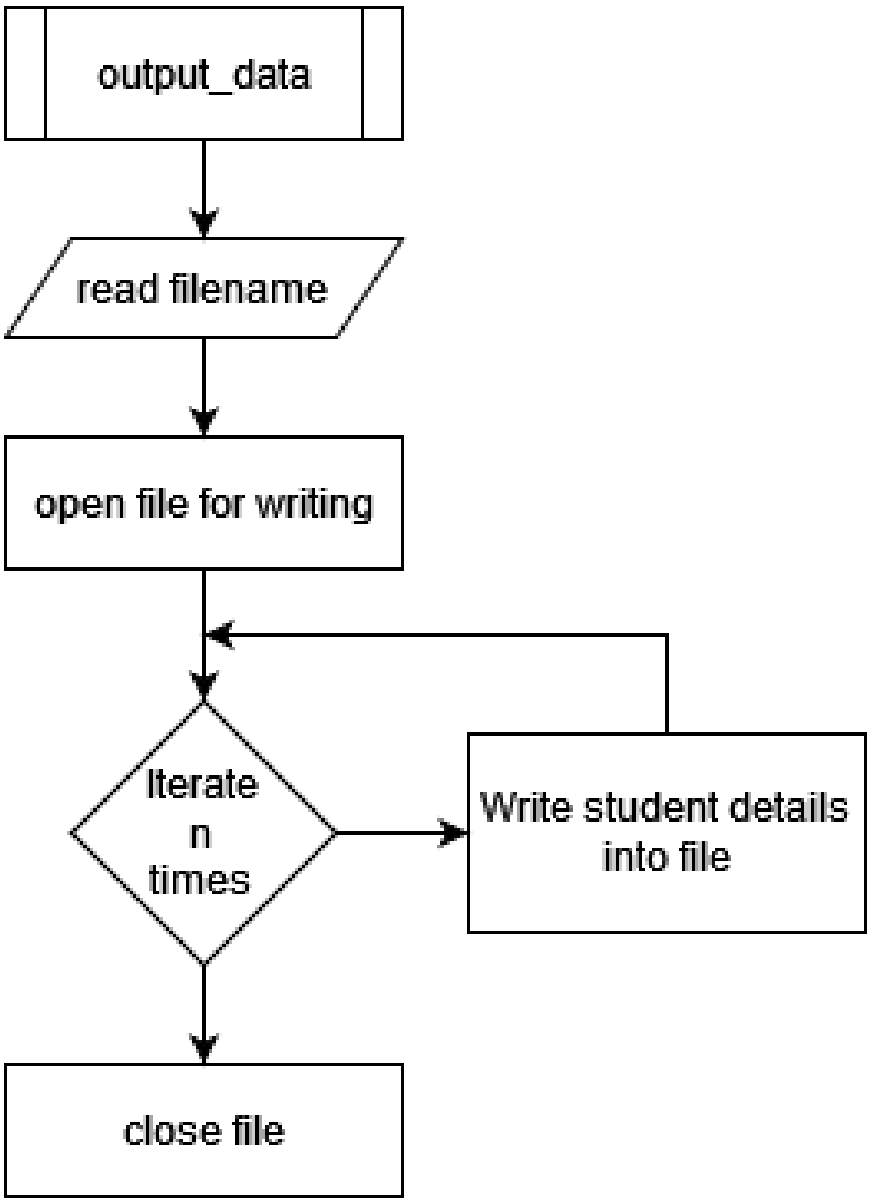
### Obtain Int Module




The obtain int module is a string into an integer

It iterates through the string and finds its length ,checks if it is a digit and adds it into a new string and converts the new string into an integer.

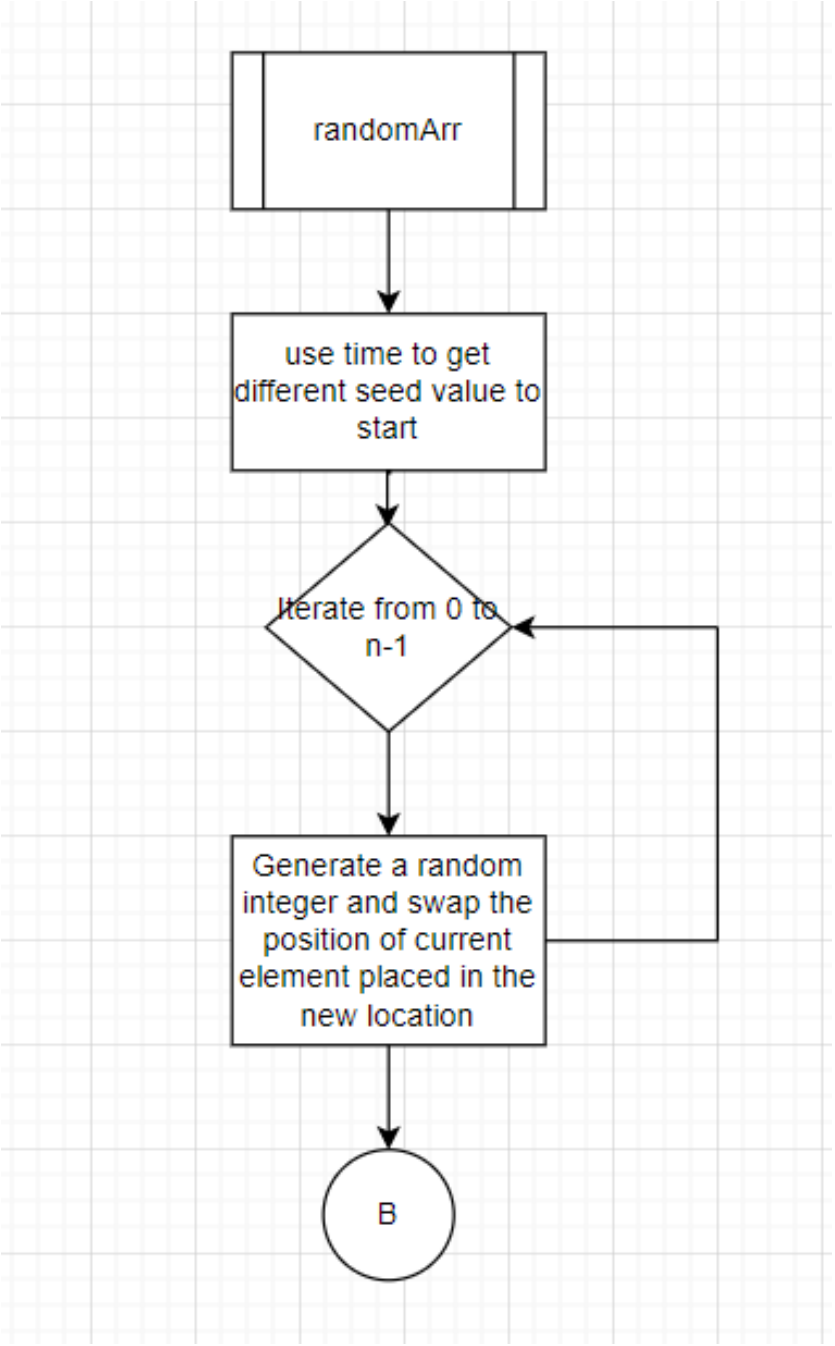
**Output Data Module**




Output Data function is used to copy the final results of seat assigning onto a file.

It opens the results file and copies all the data present in a data structure onto it.

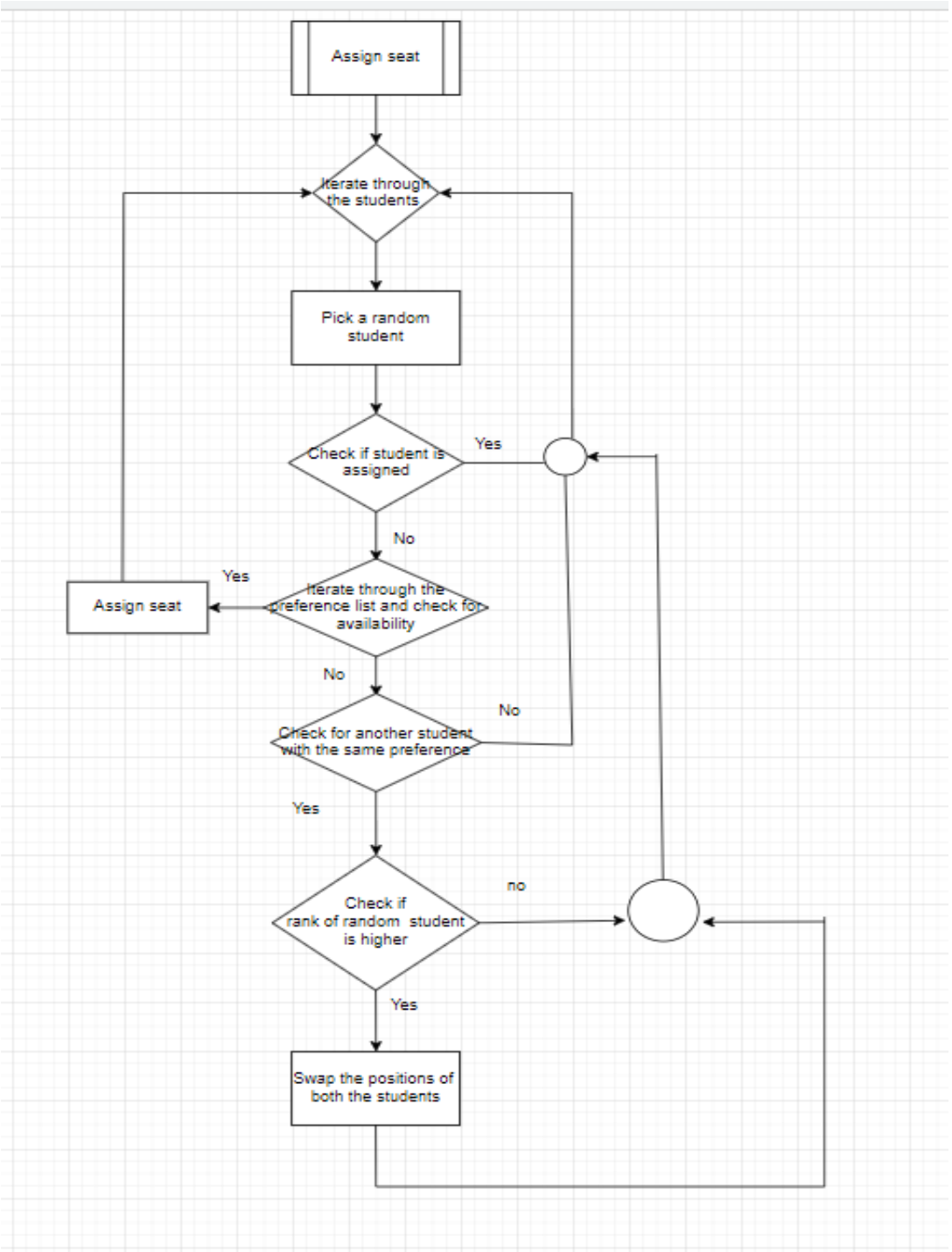
**Random Arr Module**




Random arr module is used to shuffle the data before passing to the assign seat module.Since the program uses Gale Shapley's Algorithm it is necessary for the student array to be randomised.

This module uses time function to generate a random seed value and then iterates through the students array. The random number generated is the index value for swapping the current index position.

**Assign Seat Module** Page No

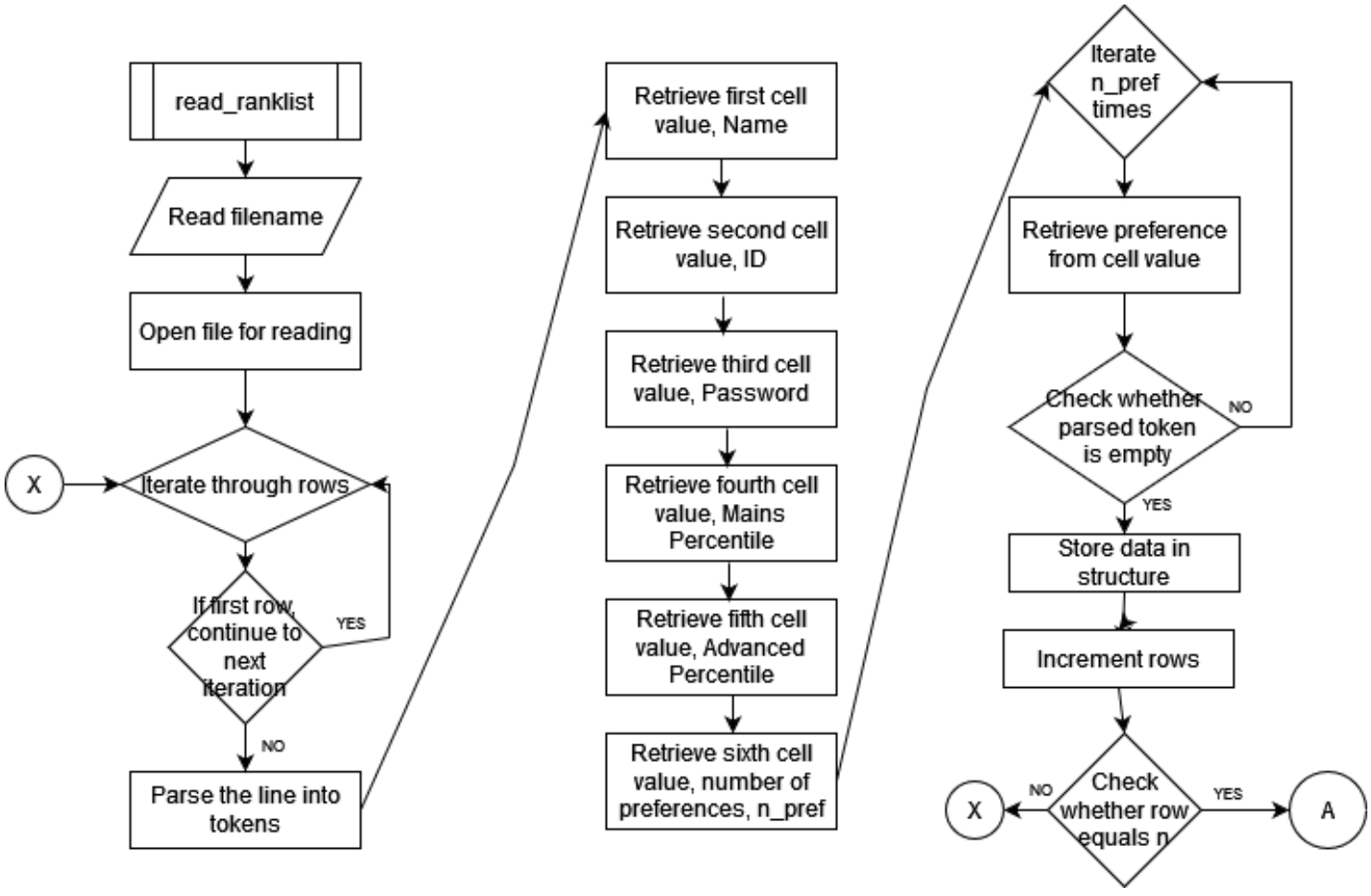



The Assign Seat module is used to allocate seats to all the students based on their preferences and their percentiles.

Initially the randomised structure is iterated and a random student is obtained.Then the module checks if the student has a seat assigned.If yes then it skips to the next iteration or else it checks for the availability of his preferences.If his preference is available then it assigns him the seat or else checks whether there

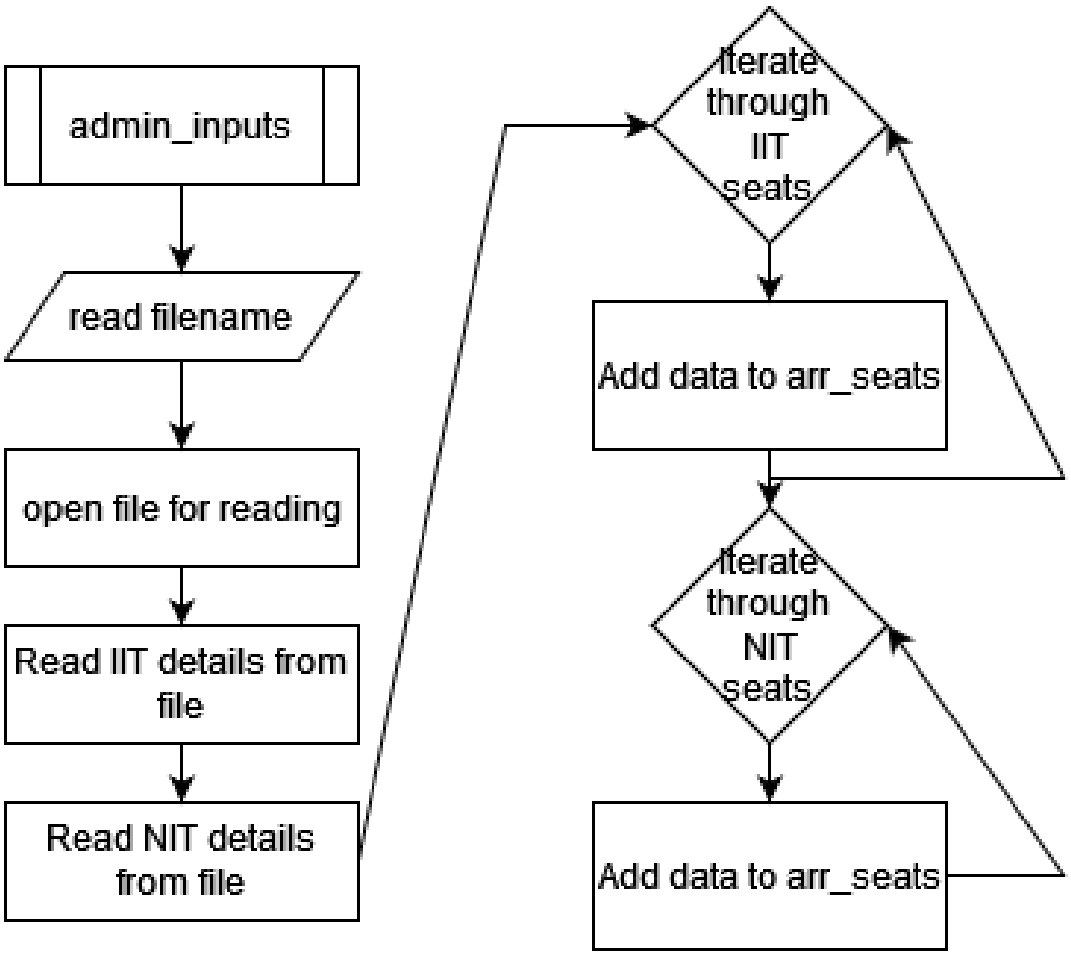
is some other person with the same preference and a lower rank who has his seat allotted.If yes then it swaps the two students or else it skips to the next iteration.

**Read Rank List Module**



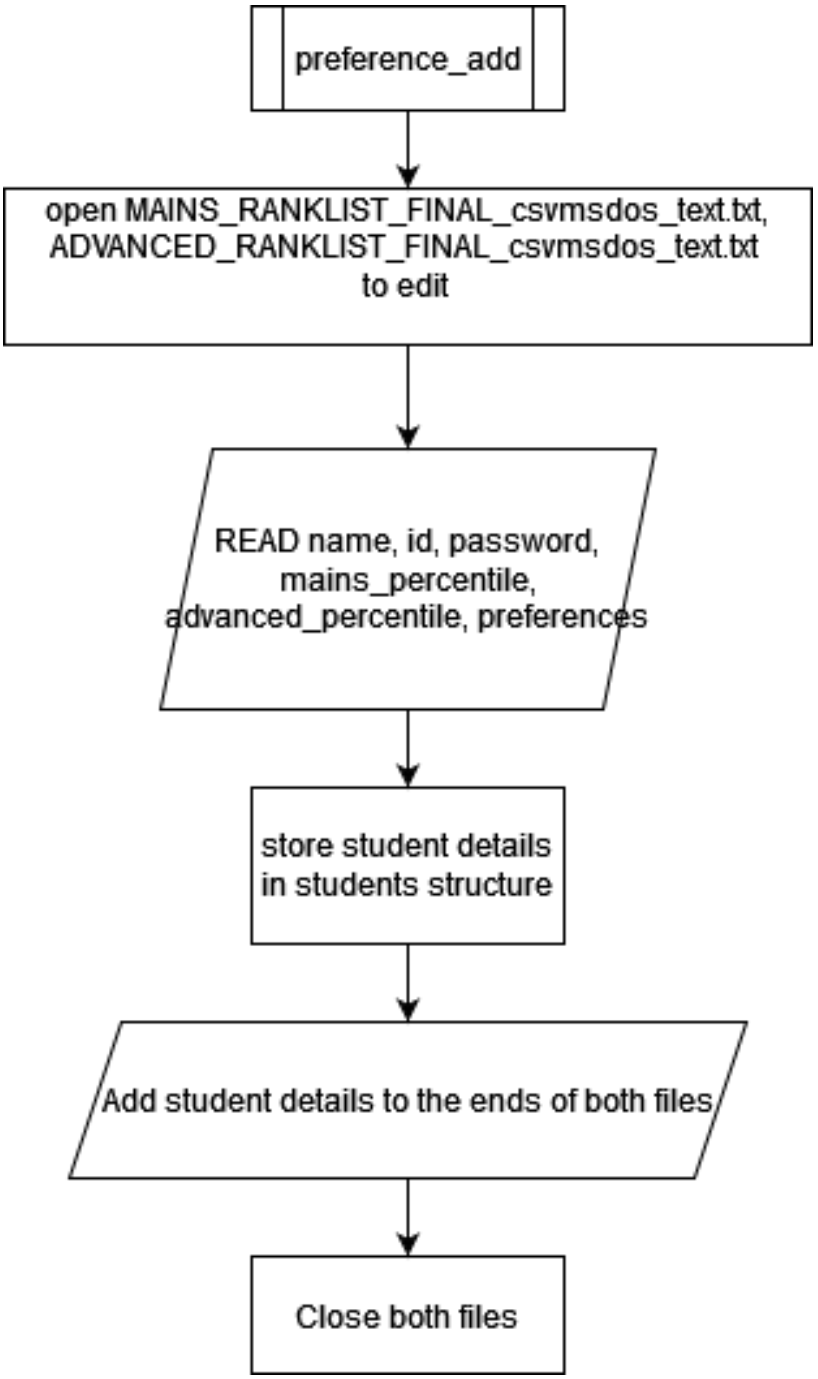

Read rank list module is used to get data from the mains and advanced rank list entered by the user and adds the data into an array of structures consisting of student details.

**Admin Inputs Module**



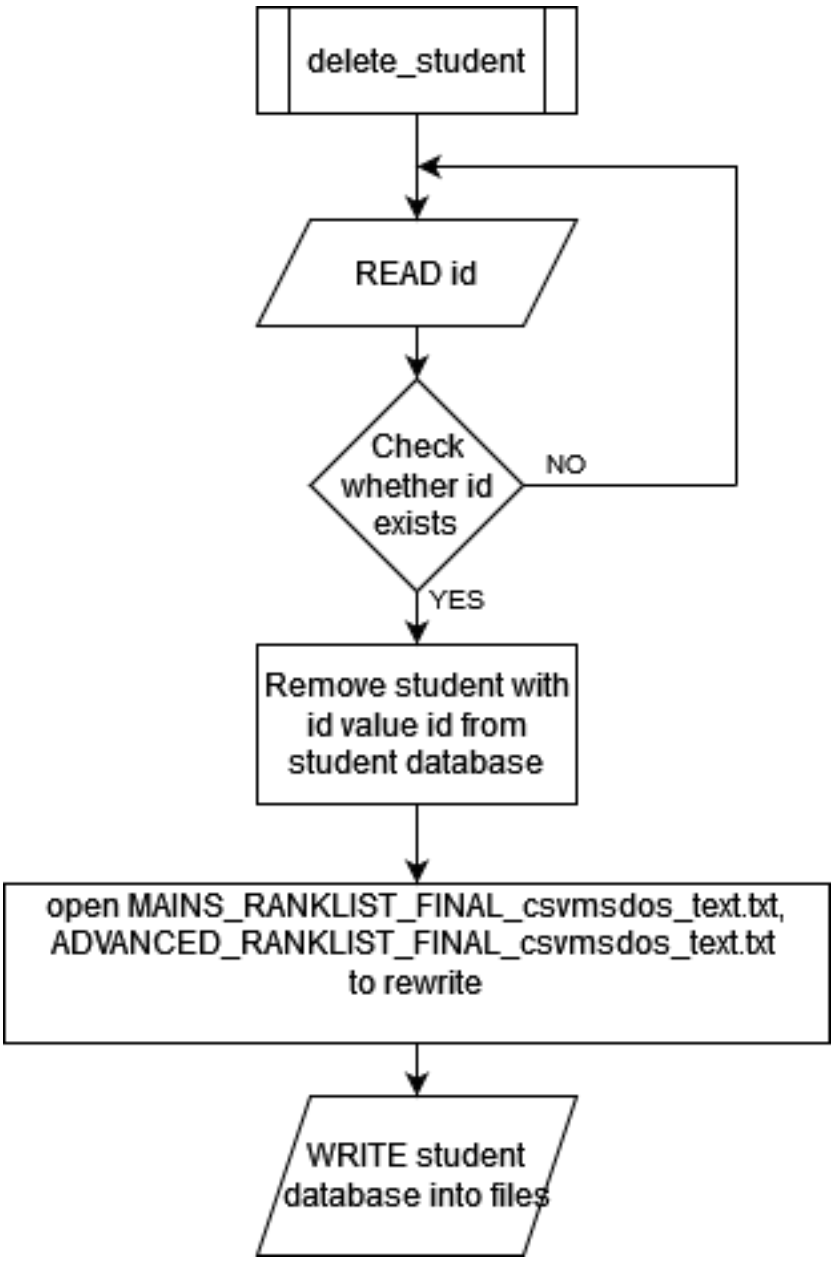

Admin inputs module is used to read the admin file and get the necessary information about the college branches,departments and number of seats.

**Adding Preference Module** Page No




preference_add module is used to read details for new student, and stores it in students database, as well as in the relevant files.

**Deleting a Student Module** Page No




delete_student module is used to input the id of the student to delete. It checks if the id exists. If it exists, the student with that id is deleted from the students database, and the files are rewritten.

**Implementation** Page No

## Organization of Data

- All the student information is stored in an array of structures ‘students’ of type student. It has attributes name, id, password, mains_percentile, advanced_percentile, n_pref, preferences, assigned_seat.
- This information is read in the read_ranklist module from either the mains or advanced rank list files.

- All the seat information is stored in an array of structures ‘arr_seats’ of type seat. It has attributes college, branch, dept, assigned_to.
- This information is read in the admin_inputs module.
- arr_mains and arr_advanced are also arrays of structures of type student. They are read from mains and advanced rank lists.

The files used are:

- MAINS_RANKLIST_FINAL_csvmsdos_text.txt
- ADVANCED_RANKLIST_FINAL_csvmsdos_text.txt
- admin_details.txt
- assigned_results.txt
- unassigned_results.txt

## Libraries used

- stdio.h - for standard input and output operations.
- string.h - for inbuilt string functions such as strcpy, strcmp, atoi, atof, etc.
- ctype.h - for inbuilt character functions such as isspace(), etc.
- time.h - for using time to change the seed for the rand() function.
- stdlib.h - for rand() function, used to shuffle students array.

## Interface Use

The only interface used to run the program was VS code Terminal.

However, to generate the random files, for the admin inputs side, Spyder (python) was also used but it is not part of the program. It is merely used to create the files for running and testing.

The original idea was to install gtk extensions to create a GUI, but due to time constraints this idea was dropped.

**Platform used for Code Development**

VS code,  version: 1.80.1 (user setup)

**Validation through Detailed Test Cases for various scenarios**

CASE #1 : high mains and high advanced rank

Student gets first preference

Ex:

| A | B | C | D | E | F | G | H | |
|---|---|---|---|---|---|---|---|---|
| Name | ID | Password | Mains Pe | Advanced | No. Prefer | Pref 1 | Pref 2 | Pref |
| Delmar | 1540 | 9TWB617 | 100 | 100 | 1 | IIT_Bombay_CSE | | |

^mains rank list (1st rank)

| A | B | C | D | E | F | G | H | |
|---|---|---|---|---|---|---|---|---|
| Name | ID | Password | Mains Pe | Advanced | No. Prefer | Pref 1 | Pref 2 | Pref |
| Delmar | 1540 | 9TWB617 | 100 | 100 | 1 | IIT_Bombay_CSE | | |
| Spenser | 1662 | MCO9U20 | 99.6 | 99.93 | 3 | IIT_Madras_IT | IIT_Madras_CSE | IIT_D |

^advanced rank list (1st rank)

TERMINAL OUTPUT:

```
Enter student ID: 1540
Enter password to login: 9TWB617
------------------------------
Logged in Successfuly
Name: Delmar
ID: 1540
Mains Percentile: 100.000000
Advanced Percentile: 100.000000
Seat details: IIT, Bombay, CSE
------------------------------
```

Student gets first preference

CASE # 2 : high mains but low advanced score, first preference is IIT

The output depends on the preference of the student

Ex:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 100 | Boy | 1690 | KMZ7L19 | 95.58 | 96.44 | 7 | NIT_Calicut_EEE |
| 101 | Willard | 1443 | QMS8B17 | 95.55 | 72.26 | 1 | IIT_Delhi_ECE |
| 102 | Margarita | 1271 | G1PLX16 | 95.52 | 72.43 | 10 | IIT_Madras_IT |

^mains rank list (100th rank)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 144 | Etti | 1641 | D4ALH13 | 99 | 72.35 | 1 | IIT_Kharagpur_EEE |
| 145 | Willard | 1443 | QMS8B17 | 95.55 | 72.26 | 1 | IIT_Delhi_ECE |

^advanced rank list (144th rank)

TERMINAL OUTPUT:

```
Enter student ID: 1443
Enter password to login: QMS8B17

------------------------------
Logged in Successfuly
Name: Willard
ID: 1443
Mains Percentile: 95.550003
Advanced Percentile: 72.260002
No seat assigned.

------------------------------
```

Here, the student has only one preference and it is IIT. The advanced mark is too low.
So no seat assigned

CASE # 3 : high mains but low advanced score, first preference is NIT

Ex:

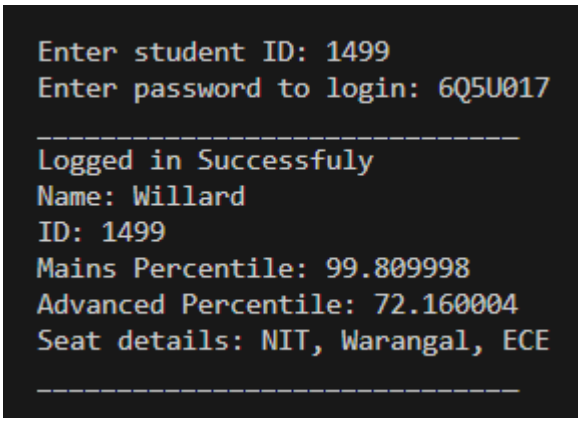
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 9 | Etan | 1491 | NKNAQ12 | 99.81 | 71.02 | 10 | IIT_Kharagpur_EEE |
| 10 | Willard | 1499 | 6Q5U017 | 99.81 | 72.16 | 1 | NIT_Warangal_ECE |
| 11 | Johan | 1910 | J6T9P10 | 99.7 | 97.22 | 1 | IIT_Bombay_EEE |

^Mains Rank list (9th rank)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 146 | Willard | 1499 | 6Q5U017 | 99.81 | 72.16 | 1 | NIT_Warangal_ECE |

^Advanced Rank list (145th rank)

TERMINAL OUTPUT:




High mains score, so the student got NIT.

CASE # 4 : high advanced but low mains mark, but first preference is NIT

Result depends on preference. If not available, no seat assigned

Ex:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 135 | Arvy | 1599 | 8MZO213 | 94 | 62.96 | 1 | NIT_Calicut_ECE |
| 136 | Wyn | 1562 | 7C92X15 | 93.98 | 95.63 | 1 | NIT_Warangal_ECE |
| 137 | Luci | 1208 | 1EOBR14 | 93.94 | 94.13 | 1 | NIT_Trichy_CSE |

^ Mains rank list (135th rank)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 26 | Eduardo | 1017 | ROGYS10 | 97.92 | 95.65 | 10 | NIT_Calicut_CSE | IIT_Khar |
| 27 | Wyn | 1562 | 7C92X15 | 93.98 | 95.63 | 1 | NIT_Warangal_ECE | |
| 28 | Egor | 1621 | RL2S119 | 92.15 | 95.3 | 2 | NIT_Warangal_EEE | IIT_Khar |

^ Advanced rank list (26th rank)

TERMINAL OUTPUT:

```
Enter student ID: 1562
Enter password to login: 7C92X15

------------------------------
Logged in Successfuly
Name: Wyn
ID: 1562
Mains Percentile: 93.980003
Advanced Percentile: 95.629997
No seat assigned.

------------------------------
```

Student only gave one preference - NIT_Warangal_ECE
Although advanced marks are high, mains marks are not high enough.
So no seat was assigned.

CASE # 5 : High advanced but low mains mark, and first preference is IIT

Ex:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 115 | Cyndie | 1215 | 2MU3U18 | 94.98 | 66.74 | 1 | IIT_Delhi_ECE |
| 116 | Hermie | 1380 | B2BCL20 | 94.93 | 92.66 | 8 | NIT_Rourkela_EEE |
| 117 | April | 1462 | MWYS215 | 94.92 | 98.26 | 1 | NIT_Warangal_CSE |
| 118 | Joyous | 1840 | F3WFN15 | 94.87 | 76.09 | 2 | IIT_Madras_CSE |

^ Mains Rank list (116th rank)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 11 | Dunstan | 1944 | SCBKX16 | 96.01 | 98.33 | 6 | IIT_Delhi_IT |
| 12 | April | 1462 | MWYS215 | 94.92 | 98.26 | 1 | NIT_Warangal_CSE |
| 13 | Cazzie | 1343 | TJ71P15 | 92.12 | 98.21 | 1 | IIT_Kharagpur_CSE |

^ Advanced rank list (11th rank)

TERMINAL OUTPUT:

```
Enter student ID: 1462
Enter password to login: MWYS215

------------------------------
Logged in Successfuly
Name: April
ID: 1462
Mains Percentile: 94.919998
Advanced Percentile: 98.260002
No seat assigned.

------------------------------
```

Here, the student's mains rank is not high enough to get an NIT seat. Her only preference was NIT_Warangal_CSE, so no seat was assigned to her.

CASE # 6 : Low mains and low advanced marks

Seat mostly not assigned, unless lucky i.e. very few people gave same preference

Ex:

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 9 | Renell | 1048 | LYISK11 | 91.62 | 73.11 | 1 | NIT_Calicut_CSE | | | |
| 0 | Henrieta | 1740 | WEEU520 | 91.62 | 67.48 | 4 | IIT_Goa_ECE | IIT_Bombay_ECE | IIT_Bombay_EEE | NIT_Calicut_EEE |
| 1 | Leupold | 1299 | QLAFV19 | 91.56 | 94.78 | 3 | NIT_Rourkela_IT | NIT_Calicut_CSE | IIT_Bombay_ECE | |

^ Mains Rank list (199th rank)

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 162 | Roddie | 1835 | X4QIQ10 | 98.01 | 67.63 | 7 | NIT_Rourkela_ECE | NIT_Warangal_CSE | NIT_Warangal_IT | NIT_Warangal_CSE |
| 163 | Henrieta | 1740 | WEEU520 | 91.62 | 67.48 | 4 | IIT_Goa_ECE | IIT_Bombay_ECE | IIT_Bombay_EEE | NIT_Calicut_EEE |
| 164 | Rori | 1133 | CQ4WI16 | 93.34 | 67.22 | 1 | IIT_Delhi_ECE | | | |

^ Advanced rank list (162nd rank)

TERMINAL OUTPUT:

```
Enter student ID: 1740
Enter password to login: WEEU520
_____________________________
Logged in Successfuly
Name: Henrieta
ID: 1740
Mains Percentile: 91.620003
Advanced Percentile: 67.480003
No seat assigned.
_____________________________
```

Both percentiles are low, so no seat assigned

CASE # 7:  special case of CASE # 4

Lucky situation: When student scores low in both mains and advanced, but still gets a seat as other students did not choose the same preference.

Ex:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 158 | Gwenneth | 1342 | GAXYO17 | 93.18 | 74.31 | 1 | IIT_Delhi_CSE | | |
| 159 | Karoly | 1996 | LF39U11 | 93.18 | 93.82 | 3 | NIT_Warangal_ECE | IIT_Madras_EEE | IIT_Delhi_EEE |
| 160 | Marnie | 1356 | VKY0V13 | 93.15 | 60.72 | 4 | IIT_Delhi_ECE | NIT_Trichy_EEE | NIT_Rourkela_IT |
| 161 | Carly | 1336 | RS6PF13 | 93.08 | 61.38 | 6 | IIT_Madras_CSE | NIT_Rourkela_IT | NIT_Calicut_ECE |

^ Mains rank list  (158th rank)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 35 | Penny | 1650 | SO2OT20 | 98.17 | 93.97 | 3 | IIT_Goa_IT | IIT_Kharagpur_EEE | IIT_Bombay_CSE |
| 36 | Laverna | 1771 | UX3NG17 | 98.62 | 93.83 | 6 | IIT_Kharagpur_CSE | NIT_Calicut_ECE | NIT_Suratkal_ECE |
| 37 | Karoly | 1996 | LF39U11 | 93.18 | 93.82 | 3 | NIT_Warangal_ECE | IIT_Madras_EEE | IIT_Delhi_EEE |
| 38 | Kym | 1332 | 4EUK216 | 96.37 | 93.75 | 6 | NIT_Warangal_ECE | IIT_Bombay_EEE | NIT_Calicut_ECE |
| 39 | Latia | 1950 | C3T1R14 | 96.95 | 93.67 | 6 | IIT_Goa_ECE | IIT_Goa_IT | IIT_Kharagpur_IT |

^ Advanced rank list  (36th rank)

TERMINAL OUTPUT:

```
Enter student ID: 1996
Enter password to login: LF39U11
_____________________________
Logged in Successfuly
Name: Karoly
ID: 1996
Mains Percentile: 93.180000
Advanced Percentile: 93.820000
Seat details: IIT, Madras, EEE
_____________________________
```

Despite this student's low advanced score, she is the student with the highest rank that chose that college. So she got the seat.

CASE # 8 : High advanced rank, but first preference is NIT

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 5 | Frankie | 1502 | CK9E415 | 93.21 | 98.88 | 5 | IIT_Bombay_EEE | IIT_Delhi_ECE | NIT_Rourkela_IT |
| 6 | Selby | 1429 | 7MMBD15 | 96.57 | 98.66 | 3 | NIT_Trichy_CSE | IIT_Kharagpur_CSE | NIT_Suratkal_ECE |
| 7 | Janaya | 1333 | US5YN20 | 98.54 | 98.65 | 8 | IIT_Bombay_ECE | NIT_Warangal_ECE | IIT_Kharagpur_CSE |

^ Advanced rank list (5th rank)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 82 | Nealon | 1173 | FO1SG11 | 96.58 | 92.86 | 3 | IIT_Delhi_EEE | NIT_Rourkela_EEE | NIT_Trichy_CSE |
| 83 | Selby | 1429 | 7MMBD15 | 96.57 | 98.66 | 3 | NIT_Trichy_CSE | IIT_Kharagpur_CSE | NIT_Suratkal_ECE |
| 84 | Gardener | 1738 | TG66D18 | 96.52 | 96.85 | 10 | IIT_Kharagpur_EEE | IIT_Goa_EEE | NIT_Warangal_EEE |

^ Mains rank list (82nd rank)

TERMINAL OUTPUT:

```
Enter student ID: 1429
Enter password to login: 7MMBD15
_____________________________
Logged in Successfuly
Name: Selby
ID: 1429
Mains Percentile: 96.570000
Advanced Percentile: 98.660004
Seat details: IIT, Kharagpur, CSE
_____________________________
```

Although the first preference was IIT, the student got NIT as his advanced score was higher.

CASE # 9 : Did not write JEE advanced, and gave only NIT preference.
[Previously encoded value by Admin]

Seat allocation is the usual process.
Ex:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 96 | Katha | 1998 | NF20S21 | 96.05 | 89.96 | 10 | IIT_Madras_EEE | I |
| 97 | Maggi | 1137 | 9CA3V11 | 96.03 | -1 | 1 | NIT_Rourkela_ECE | |
| 98 | Dunstan | 1944 | SCBKX16 | 96.01 | 98.33 | 6 | IIT_Delhi_IT | N |

^ Mains rank list (96th rank)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 197 | Sib | 1528 | HU4V918 | 96.92 | -1 | 9 | IIT_Goa_EEE | IIT_ |
| 198 | Maggi | 1137 | 9CA3V11 | 96.03 | -1 | 1 | NIT_Rourkela_ECE | |
| 199 | Vernice | 1873 | X6DRH12 | 92.49 | -1 | 6 | NIT_Rourkela_CSE | IIT_ |

^ Advanced rank list (did not write JEE advanced)

TERMINAL OUTPUT:

```
Enter student ID: 1137
Enter password to login: 9CA3V11

_____________________________
Logged in Successfuly
Name: Maggi
ID: 1137
Mains Percentile: 96.029999
Advanced Percentile: -1.000000
No seat assigned.

_____________________________
```

Unfortunately, here the student's mains percentile is not high enough to get NIT as well.
So not seat was assigned.

CASE # 10 : Did not write JEE advanced, but gave IIT preference.
[Previously encoded value by Admin]

NOTE :
For previously hard-coded values by the admin, the preference can be IIT even when the student did not write JEE advanced, i.e., advanced_percentile = -1.
In such cases, the preference is skipped while assigning seats. So only NITs are assigned.

Ex:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 18 | Bess | 1036 | 7RUZ716 | 99.27 | 77.27 | 4 | IIT_Kharagpur_EE |
| 19 | Hyatt | 1068 | 1V4T515 | 99.26 | -1 | 1 | IIT_Delhi_EEE |
| 20 | Raimund | 1027 | 10TYI14 | 99.22 | 97.02 | 2 | IIT_Delhi_EEE |

^ Mains rank list (18th rank)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 195 | Marin | 1430 | M72P719 | 99.36 | 60.09 | 5 | IIT_Goa_EEE |
| 196 | Hyatt | 1068 | 1V4T515 | 99.26 | -1 | 1 | IIT_Delhi_EEE |
| 197 | Sib | 1528 | HU4V918 | 96.92 | -1 | 9 | IIT_Goa_EEE |

^ Advanced rank list (did not write JEE advanced)

TERMINAL OUTPUT:

```
Enter student ID: 1068
Enter password to login: 1V4T515

------------------------------
Logged in Successfuly
Name: Hyatt
ID: 1068
Mains Percentile: 99.260002
Advanced Percentile: -1.000000
No seat assigned.

------------------------------
```

Here, the only preference is IIT, but the student did not write JEE advanced.

NOTE:
Since this is a previously-encoded value from the admin, validation of input is not provided. To ensure merit, the IIT preferences are skipped during the iterations.

CASE # 11 : Did not write JEE advanced, but gave IIT preference.

[User-entered value]

```
Enter 1(Admin) or 2(student login) or 3(add new student) or 4(update seat allocation results) or 5(Delete a student) or 0(to terminate): 3


Enter student name: shyam
Enter student ID: 9999
Enter the password: hello1234
Enter student's mains percentile: 48.96
Enter the student's advanced percentile: -1
Enter the no.of preferences: 1
Enter 1 preference: IIT_Madras_IT

You did not write JEE advanced, but gave preference as IIT_Madras_IT. You can only enter NIT preferences.
Enter preference : NIT_Trichy_EEE

Enter 1(Admin) or 2(student login) or 3(add new student) or 4(update seat allocation results) or 5(Delete a student) or 0(to terminate): 4
The seats are successfully allocated!
To see all results, you need admin access (1).
Otherwise, you can login as a student (2) to see your results

Enter 1(Admin) or 2(student login) or 3(add new student) or 4(update seat allocation results) or 5(Delete a student) or 0(to terminate): 2


Enter student ID: 9999
Enter password to login: hello1234
------------------------------
Logged in Successfuly
Name: shyam
ID: 9999
Mains Percentile: 48.959999
Advanced Percentile: -1.000000
No seat assigned.
------------------------------

Enter 1(Admin) or 2(student login) or 3(add new student) or 4(update seat allocation results) or 5(Delete a student) or 0(to terminate): 0


TERMINATING SYSTEM...
```

In the above code execution, the user is not allowed to enter an IIT preference if he did not write JEE advanced.

**Limitations**

- The store string values, strcpy() from string.h header file, is used. So realloc() cannot be used as it creates a pointer which acts as an array. So, the size of the students array cannot be dynamically initialized. Due to this, we hardcoded the size of students[] array.

- Here, ‘students’ is an array of structures containing all student information. arr_mains and arr_advanced are mains and advanced rank lists respectively. As all the information is already available in the students array, it is almost useless to use these 2 rank lists.

- No GUI was created. GUIs are implemented by installing gtk extension. Due to time constraints, the program was not implemented using GUI.

- The terminating condition is done by using a flag. The flag’s value changes if there is a change in seat allocation for n students. If the flag changes, the loop is terminated. Sometimes, the only change might be in the first few students. But the program will still iterate through all the other students, causing the time complexity to be high.

- The maximum number of preferences for 250 students is only 10. In practical scenario, it is 31. Due to lack of RAM space, only 10 preferences could be implemented

## Observations from the Societal, Legal, Environmental and Ethical perspectives

### Societal Perspective

Most qualified and deserving individuals are selected, fostering a competitive and skilled workforce or student body. However, there are also potential societal drawbacks to consider. The intense competition inherent in rank-based seat allotment processes can create a high-pressure environment, leading to stress, anxiety, and unhealthy competition among candidates. This can have negative effects on mental health and well-being, especially for individuals who may not secure their desired seats despite their efforts.

Additionally, rank-based seat allotment processes may inadvertently contribute to societal inequalities. Candidates from disadvantaged backgrounds, who may face various socio-economic challenges, may have limited access to resources or educational opportunities that could impact their rankings. This can result in a lack of diversity and perpetuate existing social inequities within the education and job sectors.

### Legal Perspective

From a legal standpoint,Our rank-based seat allotment systems adhere to principles of fairness, transparency, and equal opportunity.

The process is designed in a way that ensures all candidates are treated fairly, and no discrimination based on factors such as gender, race, or disability occurs. Legal frameworks also include the provision of reservation or quota systems to address historical disadvantages faced by certain groups.

### Environmental Perspectives

From an environmental perspective,Our rank-based seat allotment systems can impact sustainability in terms of transportation and resource utilization. For example, if candidates are required to physically attend counseling sessions or allocation procedures, it can result in increased travel and carbon emissions. Implementing online systems and virtual counseling can help reduce the environmental footprint of the process

.Additionally, the infrastructure and resources required to accommodate a large number of candidates during the seat allotment process may have environmental implications. This includes the construction and maintenance of examination

centers, counseling venues, and other facilities.

**Learning Outcomes**

- We thoroughly understood the working principle of the Gale Shapley Algorithm, and implemented it in our code.
- We learnt how to use the C programming language to develop a properly working software.
- We learnt various inbuilt C functions such as atoi, atof, strtok, rand, srand.
- Understood how to access files, structures and arrays complementary to each other.
- We also learn several small, yet important, tips in C that help in the future. Ex: using getchar() between integer and string input to avoid skipping the string input.
- We learnt how to efficiently work by simultaneously doing the work and clarifying eachother's doubts.
- We learnt how to work as a group and divide the work equally amongst ourselves.

## References


- https://www.geeksforgeeks.org/stable-marriage-problem/
- https://iq.opengenus.org/gale-shapley-algorithm/