Problem 2 Gale-Shapley Algorithm with slight modification:

Initialize number of students as 5000 and colleges as 15

Create a 5000x15 matrix for preferences of colleges of all the 5000 students.

Create a 15x5000 matrix for preferences of students of all the 15 colleges.

Create a 1x15 list which would have the number of intake of a college

Create a 15x(number of intakes of every college for each row) list which would indicate all the allocated students to the given college.

Find number of students who wont be selected by a college (Total students -sum of number of seats in all colleges).

Create a free students list which contains students who are looking for a college that prefers them. Fill it with all the students at the start.

Loop through the free students,

Check the students preferences(Loop through them)

if the preferred college is not filled

add the student in this college and break the preferences loop

if the preferred college is filled

check the colleges preference list

if a student less preferred is already allocated in college

replace less preferred with our new student

add less preferred student in the free student list

break till the preferences loop

if the student is not getting added to any college after these loops, it means there already are students more preferred than him in the colleges. Hence, he would be surely rejected. So he is added in the rejected_from_all list.

Keep on looping this whole pseudo code mentioned above until the exit condition.
Exit condition: When the length reject_from_all list is equal to the number of students who wont be selected.

Why the program will always terminate?

The program ensures that each student has a chance to check all the colleges if he can be selected in place of a less preferred student. After checking this, the student is added to reject_from_all list. Eventually this list would contain all those students who are less preferred which would eventually be equal to number of students who do not get selected. Hence the program will always terminate.