Algorithms of

Automated Processes

Center Allotment – a 2-staged process

STAGE 1: Allotment on Prioritized Availability

```
for(each student of student linked-list)
{       for(each center choice of student)
        {       while(center != center choice of student)
                        Traverse center linked-list; //i.e. reach a center chosen by the student.
                if(center is not fully occupied)
                {       Allot that center to student;
                        Proceed to next student;
                }
        }
        if(center of choice could not be alloted to the student)
                Turn flag on; // flag signifies that there is such a student who has not been alloted a center.
}
```

STAGE 2: Allotment of remaining students (who could not be alloted their chosen centers)

```
if(flag is on)  //i.e. there is such a student who has not been alloted a center.
{       for(each student of student linked-list)
        {       if(student has not been alloted seat)
                {       while(center is fully occupied)
                                Traverse center linked-list; //i.e. reach a center which has not been fully occupied.
                        Allot that center to the student.
                }
        }
}
```

Seat Allotment

```
Sort the student linked list according to rank; // call the sortByRank() function.

for(each student of student linked-list)
{
    if(student has appeared for counceling) //i.e. if the student has entered his branch choice.
    {
        for(each branch choice of student)
        {
            while(branch != branch choice of student)
                Traverse branch linked-list; //i.e. reach a branch chosen by the student.
            if(branch seat[reservation category of student] is not full)
            {
                Allot that branch to student;
                Proceed to next student;
            }
        }
    }
}
```

Other Snippets

Password Character & checking

```
Let correct password be stored in string password

Declare char array passwordEntered[ ]; // to store password entered by user
Initialise passwordCounter = 0;

while(input character != '\r') //i.e. until the user presses enter.
{
    Read one character from user. //getch() function is used, which does not display the character typed.
    if(input character is alphanumeric)
    {
        passwordEntered[passwordCounter] = input character;
        Increment passwordCharacter by 1;
        Print '*';
    }
}

passwordEnterd[passwordCounter] = '\0'; // add null termination.

Compare passwordEntered and password, to check if passwordEntered is correct;
```

Roll Number & ID Generating

This process is same for generating (i) Roll number for new student,
(ii) ID of center or branch.

```
// While Preparing a new node to store information of new node.

if(student linked-list is empty)
{
    Assign roll number as 1;
    Connect the new node;
}
else
{
    Traverse to last node of student linked-list;
    Assign roll number of new student = roll number of the last node + 1;
    Connect new node at end;
}
```