At all, this algorithm program has following data structure. We have vector available[ ] , which indicates the number of available resources, matrices maximum[][], which defines the maximum demand of each process, allocation[][],which defines the number of resources of each type currently allocated to each process and matrix need[][] ,which indicates the remaining resource need of each process and calculated by subtracting from maximum number of instances number of allocated resources of each type for each process. In the beginning user must declare the maximum number of instances of each resource type that all customers(processes) need and the number of resources of each type currently allocated to each process . Then we check whether our maximum number not exceeds the total number of resources(vector all[]) in the system. If not, we check whether available resource instances are bigger than needed, if so, for output safe sequence massive we write number of process then to available resources we add its preciously allocated resources as this process will be executed. And this will be continued within in loop until all process will be executed. If our system as a result is in safe state we output our safe sequence. So, result is shown below:

