# ASSIGNMENT 2

**Submission Date : 12/11/2021**
**Due Date : 21/11/2021 23:59**
**TA :** Necva BÖLÜCÜ

Accept your *Click here to accept your Assignment 2*.

**Instructions:** This assignment consists of two parts. The first part involves a series of theoretical questions and the second part involves coding. The goal of this assignment is to make you understand and familiarize with deadlock and allocate resources.

## Part I: Theory Questions

1. Name the four conditions required for deadlock and give a brief description of the conditions.

    - first, many resources are used by a single process ; no simultaneous use and no sharing (mutual exclusion)
    - second the process requests another resource without releasing the resource it has; moreover, it can request the source held by another process.(hold and wait)
    - third resources are not prefetched to avoid deadlocks.. Allocates resources to processors for the duration of the work until the task is finished. Therefore, there is no temporary reallocation of resources.(no preemption)
    - fourth Two processes refuse to withdraw, request the previous resource, and wait for each other to withdraw. a queue is formed (circular wait)

2. What is the difference between deadlock prevention and deadlock avoidance?

    - The problem can be eliminated by eliminating at least one of the above 4 conditions causing deadlock prevention. These are valid as requested and given in the system.(deadlock prevention )
    - If a request is made for a resource , the resources are looked up . A deadlock situation is when the resource is granted if the future resource process needs resources that are out of the box. source should not be given as much as possible, if not possible, should be given (deadlock avoidance)
    - as a result, one ensures that at least one of the conditions necessary to cause a deadlock never occurs, while the other ensures that the anti-lock system does not enter an unsafe state.

3. What category does Bankers algorithm falls in and why?

    Deadlock avoidance because ;

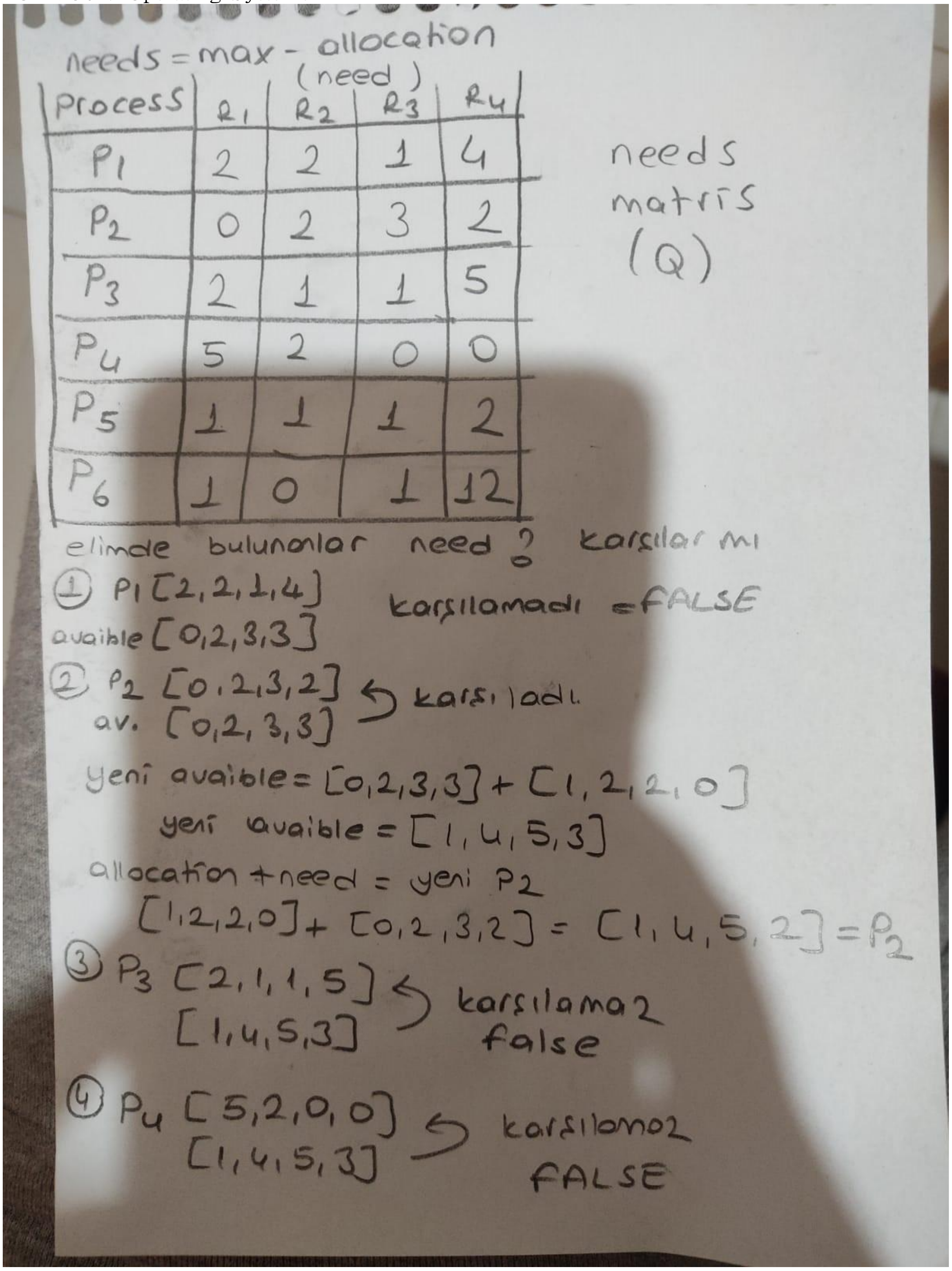
    It has been defined above that if a request is made for a resource, resources are sought. that is, it performs a security test for predetermined values, for example;

    there are people who have many accounts in banks. If we have to give a loan from the bank, the sum of the money in these accounts subtracts the loan amount from the bank's total money. Then it checks if the difference is greater than the money in the bank. The reason is that even if people with all accounts withdraw their money at once, the bank will have enough money.

4. A system that uses the Banker' s Algorithm has six processes and four types of reusable resources (R1, R2, R3 and R4). The current allocation, maximum needs (requests) andavailable resources are as shown below.

Calculate the Needs matrix (Q) and then determine if this is a safe state? If so, give a sequence in which the processes can be serviced. If it is not a safe state then list the processes which are deadlocked. Show all your work and the content of available vector in each step clearly.

| | Allocation Matrix (A) | | | | Request Matrix (C) | | | | Available Matrix (V) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Process | R1 | R2 | R3 | R4 | R1 | R2 | R3 | R4 | R1 | R2 | R3 | R4 |
| P1 | 1 | 0 | 0 | 0 | 3 | 2 | 1 | 4 | 0 | 2 | 3 | 3 |
| P2 | 1 | 2 | 2 | 0 | 1 | 4 | 5 | 2 | | | | |
| P3 | 2 | 1 | 3 | 0 | 4 | 2 | 4 | 5 | | | | |
| P4 | 1 | 4 | 1 | 0 | 6 | 6 | 1 | 0 | | | | |
| P5 | 1 | 2 | 0 | 1 | 2 | 3 | 1 | 3 | | | | |
| P6 | 1 | 4 | 3 | 1 | 2 | 4 | 4 | 13 | | | | |



needs = max - allocation

(need)

| Process | R1 | R2 | R3 | R4 |
|---|---|---|---|---|
| P1 | 2 | 2 | 1 | 4 |
| P2 | 0 | 2 | 3 | 2 |
| P3 | 2 | 1 | 1 | 5 |
| P4 | 5 | 2 | 0 | 0 |
| P5 | 1 | 1 | 1 | 2 |
| P6 | 1 | 0 | 1 | 12 |

needs matris (Q)

elimde bulunanlar need ? karsılar mı

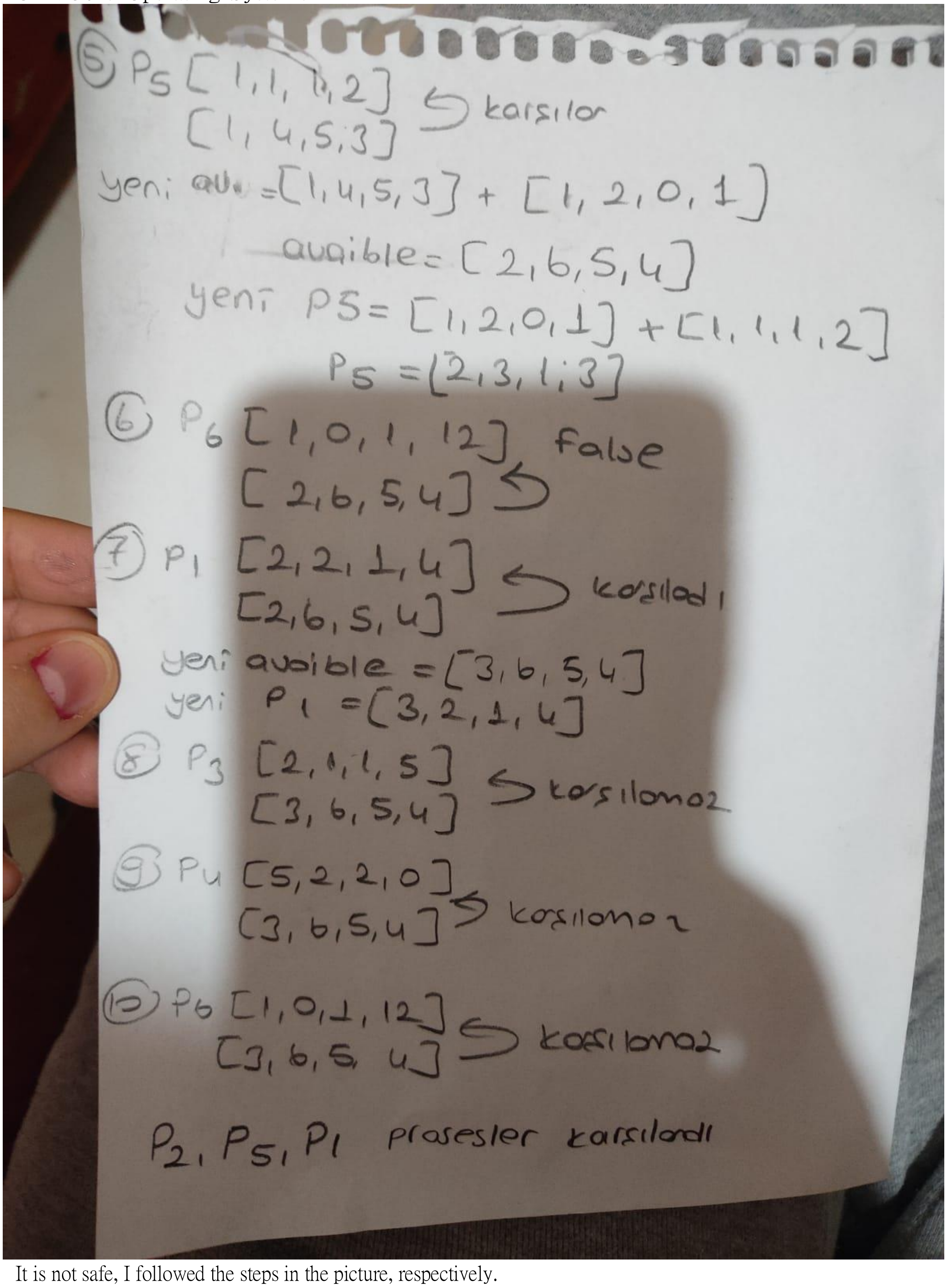
① P1 [2,2,1,4] karsılamadı = FALSE
avaible [0,2,3,3]

② P2 [0,2,3,2] karsıladı.
av. [0,2,3,3]

yeni avaible = [0,2,3,3] + [1,2,2,0]
yeni avaible = [1,4,5,3]
allocation + need = yeni P2
[1,2,2,0] + [0,2,3,2] = [1,4,5,2] = P2

③ P3 [2,1,1,5] karsılamaz
[1,4,5,3] false

④ P4 [5,2,0,0] karsılamaz
[1,4,5,3] FALSE



5) P5 [1,1,1,2] ↩ karsılar
[1,4,5,3]
yeni av. =[1,4,5,3] + [1,2,0,1]
avaible= [2,6,5,4]
yeni P5= [1,2,0,1] + [1,1,1,2]
P5 =[2,3,1,3]

6) P6 [1,0,1,12] false
[2,6,5,4] ↩

7) P1 [2,2,1,4] ↩ karsıladı
[2,6,5,4]
yeni avaible =[3,6,5,4]
yeni P1 =[3,2,1,4]

8) P3 [2,1,1,5] ↩ karsılamaz
[3,6,5,4]

9) P4 [5,2,2,0] ↩ karsılamaz
[3,6,5,4]

10) P6 [1,0,1,12] ↩ karsılamaz
[3,6,5,4]

P2, P5, P1 prosesler karsılandı

It is not safe, I followed the steps in the picture, respectively.
p2 , p5 , p1 are welcomed. but p3, p4 p6 is not met. p3 p4 p6 processes cannot be run safely with the amount of resources idle. deadlock can occur if the system is in an insecure state

## Part II:Banker’s Algorithm

Banker’ s algorithm is a deadlock avoidance algorithm. It is run by the operating system whenever a process requests resources.  It is named so because this algorithm is used in banking systems to determine whether a loan can be granted or not. Whenever a new process is created, it must specify the maximum instances of each resource type that it needs, exactly.

Banker’ s algorithm are characteristics:

- If any process requests for a resource, then it has to wait.

- This algorithm consists of advanced features for maximum resource allocation.
- There are limited resources in the system we have.
- In this algorithm, if any process gets all the needed resources, then it is that it should return the resources in a restricted period.
- Various resources are maintained in this algorithm that can fulfill the needs of at least one client.

Algorithm for checking to see if a state is safe:

1. Calculate the need matrix, and available resources $\boldsymbol{A}$.
2. Look for row, R, whose unmet resource needs all $\leq \boldsymbol{A}$. If no such row exists, system will eventually deadlock since no process can run to completion
3. Assume process of row chosen requests all resources it needs and finishes. Mark process as terminated, add all its resources to the A vector.
4. Repeat steps 1 and 2 until either all processes marked terminated (initial state was safe) or no process left whose resource needs can be met (there is a deadlock).

```
Input:
4 // # of processes
5 // # of resources
//Max resource matrix
1 1 2 1 3
2 2 2 1 0
2 1 3 1 0
1 1 2 2 1
//resource allocation matrix
1 0 2 1 1
2 0 1 1 0
1 1 0 1 0
1 1 1 1 0
//resource available
0 0 2 1 2
```

---

```
Output:
Need Matrix :
0 1 0 0 2
0 2 1 0 0
1 0 3 0 0
0 0 1 1 1
Safe sequence is :
D A C B
Change in available resource matrix :
1 1 3 2 2
2 1 5 3 3
3 2 5 4 3
```

```
5 2 6 5 3
```

**Submit** You are required to submit all of your code along with a report in PDF format. The codes you submit should be well commented. The report should contain answers of the theoretical questions and the results of your code with a discussion.

The ZIP file will be submitted via Github Classroom. *Click here to accept your Assignment 2*

## 1 Grading Policy

1. Code (75): Reading input file: 2, Constructing max resource matrix and resource allocation matrix: 3, Calculation of available resources A: 5, Implementing Baker' s Algorithm: 60, Writing the expected output to output file: 5
2. Report (25): Theory part: 25

## Important Notes

- Do not miss the submission deadline.
- Compile your code before submitting your work to make sure it compiles without any problems.
- Save all your work until the assignment is graded.
- All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudocode) will not be tolerated. In short, turning in someone else' s work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.
- You may assume that the input files will be given as:
  - Assignment2.java input_file output_file or Assignment2.c input_file output_file