

Student Name: Bishwajyoti Roy

Student ID: 11810516

Email Address: bishwajyoti456@gmail.com

GitHub Link: <https://github.com/BRoy98/OSLab>

Code: Banker's Algorithm

Description: The Banker algorithm is a deadlock avoidance algorithm that tests for safety by simulating the allocation of predetermined maximum possible amounts of all resources, and then makes an "s-state" check to test for possible deadlock conditions for all other pending activities, before deciding whether allocation should be allowed to continue.

My code takes the input values dynamically and calculates them instead of calculating pre defined values.

Code snippet:

```
// calculate the need of all the process
void calculateNeed(int need[processCount][resourcesCount], int
max[processCount][resourcesCount],
    int alloc[processCount][resourcesCount])
{
    for (int i = 0; i < processCount; i++)
        for (int j = 0; j < resourcesCount; j++)
            // need = max - allocation
            need[i][j] = max[i][j] - alloc[i][j];
}

int checkSafe(
    int alloc[processCount][resourcesCount],
    int max[processCount][resourcesCount],
    int avvail[resourcesCount])
{
    int need[processCount][resourcesCount];
    calculateNeed(need, max, alloc);
    bool finish[processCount];
```

```

int work[resourcesCount];
int sequence[processCount];
int finishCount = 0;

for (int i = 0; i < resourcesCount; i++)
    work[i] = avail[i];

while (finishCount < processCount)
{
    int found = 0;
    for (int p = 0; p < processCount; p++)
    {
        if (finish[p] == 0)
        {
            int j;
            int temp = 1;
            for (j = 0; j < resourcesCount; j++)
                if (need[p][j] > work[j])
                    break;

            if (j == resourcesCount)
            {
                for (int k = 0; k < resourcesCount; k++)
                    work[k] += alloc[p][k];

                sequence[finishCount++] = p;

                finish[p] = 1;

                found = true;
            }
        }
    }

    if (found == false)
    {
        return false;
    }
}

// // printTitle();
// for (int i = 0; i < processCount; i++)
//     printf("%d ", sequence[i]);

```

```
    return true;
}
```

Boundary Conditions:

1. My code supports up to 5 process and up to 4 resource values.
2. Input values can not be edited later.

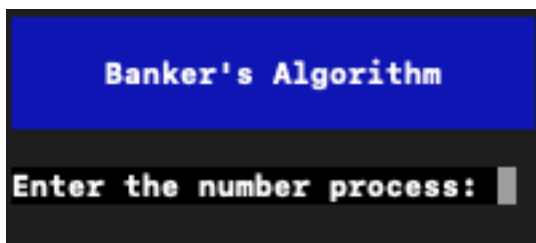
Test Cases:

Input:

Available				Processes	Allocation				Max			
A	B	C	D		A	B	C	D	A	B	C	D
1	5	2	0	P0	0	0	1	2	0	0	1	2
				P1	1	0	0	0	1	7	5	0
				P2	1	3	5	4	2	3	5	6
				P3	0	6	3	2	0	6	5	2
				P4	0	0	1	4	0	6	5	6

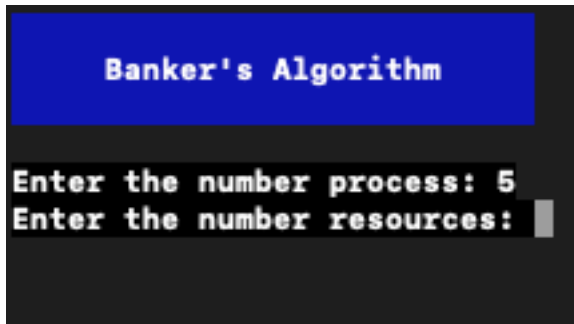
Step 1: It asks for number of process we have.

Input: 5



Step 2: It asks for number of resource we have.

Input: 4



A screenshot of a terminal window titled "Banker's Algorithm". It contains two lines of text: "Enter the number process: 5" and "Enter the number resources: " followed by a cursor.

Step 3: It asks for all the resource values for all process. It updates dynamically after every user input




A screenshot of a terminal window titled "Banker's Algorithm" with a red header "Allocation". It prompts for "Enter Process 0 (P0) Values" and lists resources A, B, C, and D with values 0, 0, 1, and 2 respectively, each followed by a cursor.



A screenshot of a terminal window titled "Banker's Algorithm" with a red header "Allocation". It prompts for "Enter Process 1 (P1) Values" and lists resources A, B, C, and D with values 1, 0, 0, and 0 respectively, each followed by a cursor.

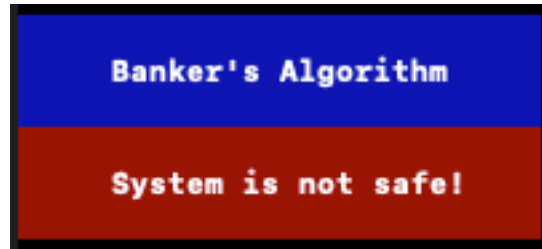
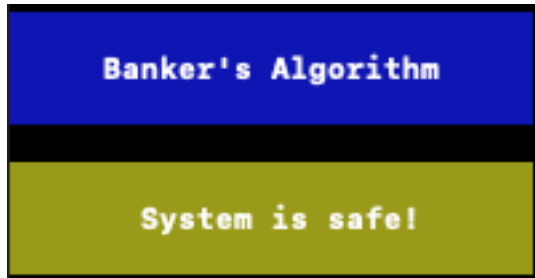


A screenshot of a terminal window titled "Banker's Algorithm" with a red header "MAX". It prompts for "Enter Process 3 (P3) Values" and lists resources A, B, C, and D with values 1, 0, 9, and 5 respectively, each followed by a cursor.



A screenshot of a terminal window titled "Banker's Algorithm" with a red header "Available". It lists resources A, B, C, and D with values 2, 4, 3, and 6 respectively, each followed by a cursor.

Output: After all the input, it should show the system status like this:



Github Link: <https://github.com/BRoy98/OSLab/blob/master/bankers-algo.c>