Name: Saqib Shehzad

Registration No. : 2021-CS-187

Submitted to Mam Abqa Javed

OS Lab Mid Paper.

Task:

Code:

```
#include <stdio.h>

// Maximum number of accounts and resources
#define MAX_ACCOUNTS 10
#define MAX_RESOURCES 5

// Available resources (G, Fixed deposit, Monthly Income Scheme, Gold, etc.)
int available[MAX_RESOURCES];

// Maximum resources that can be allocated to each account
int maximum[MAX_ACCOUNTS][MAX_RESOURCES];

// Resources currently allocated to each account
int allocation[MAX_ACCOUNTS][MAX_RESOURCES];

// Remaining resources needed by each account
int need[MAX_ACCOUNTS][MAX_RESOURCES];

// Array to track finished accounts
int finished[MAX_ACCOUNTS];
// Number of accounts and resources
int num_accounts, num_resources;
```

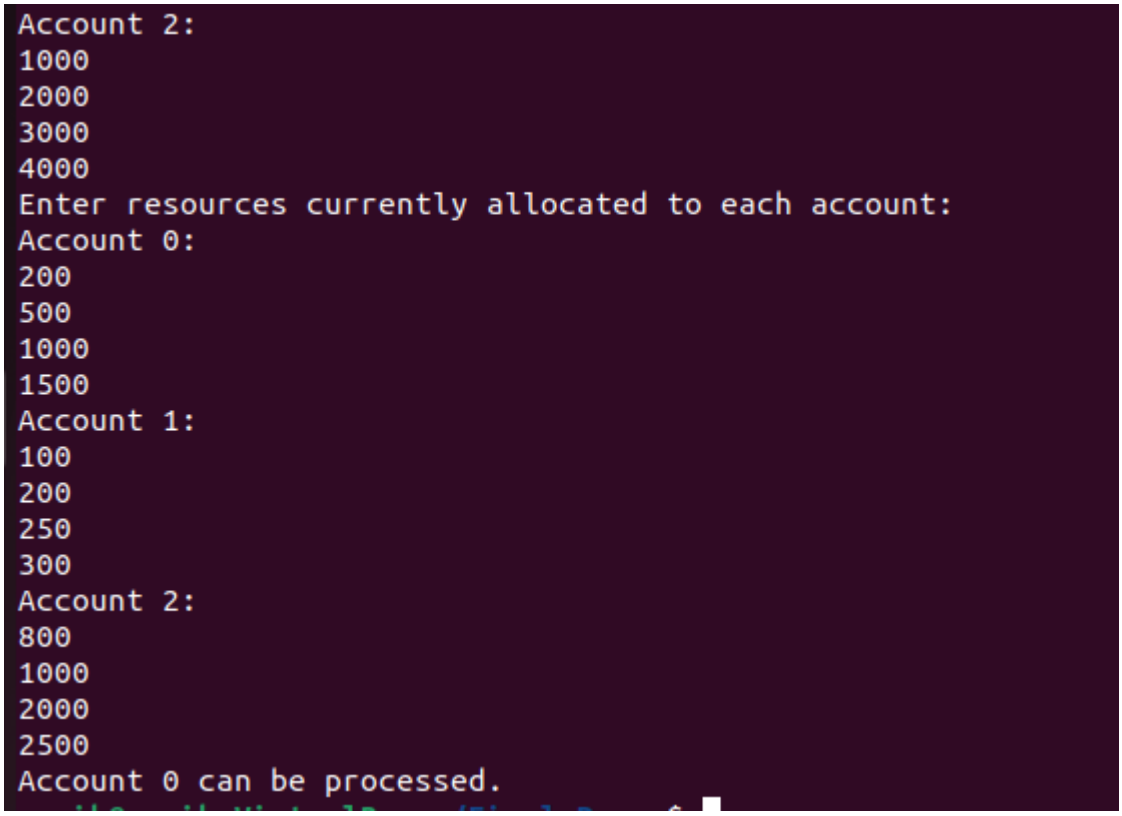
```
// Function to check if an account can be processed
int is_safe(int account)
{
    int i, j;
    int work[MAX_RESOURCES];
    // Initialize work array
    for (i = 0; i < num_resources; i++) {
        work[i] = available[i];
    }
    // Check if account can be finished
    for (i = 0; i < num_resources; i++) {
        if (need[account][i] > work[i]) {
            return 0;
        }
    }
    // Simulate allocation of resources to account
    for (i = 0; i < num_resources; i++) {
        work[i] += allocation[account][i];
    }
    // Check if all accounts can be finished
    for (i = 0; i < num_accounts; i++) {
        if (!finished[i]) {
            for (j = 0; j < num_resources; j++) {
                if (need[i][j] > work[j]) {
                    break;
                }
            }
            if (j == num_resources) {
                finished[i] = 1;
                for (j = 0; j < num_resources; j++) {
                    work[j] += allocation[i][j];
```

```
            }
        }
    }
    // Check if account can be finished
    if (finished[account]) {
        return 1;
    } else {
        return 0;
    }
}
int main()
{
    int i, j;
    // Input number of accounts and resources
    printf("Enter number of accounts: ");
    scanf("%d", &num_accounts);
    printf("Enter number of resources: ");
    scanf("%d", &num_resources);
    // Input available resources
    printf("Enter available resources:\n");
    for (i = 0; i < num_resources; i++) {
        scanf("%d", &available[i]);
    }
    // Input maximum resources for each account
    printf("Enter maximum resources for each account:\n");
    for (i = 0; i < num_accounts; i++) {
        printf("Account %d:\n", i);
        for (j = 0; j < num_resources; j++) {
            scanf("%d", &maximum[i][j]);
        }
```

```
    }
    // Input resources currently allocated to each account
    printf("Enter resources currently allocated to each account:\n");
    for (i = 0; i < num_accounts; i++) {
        printf("Account %d:\n", i);
        for (j = 0; j < num_resources; j++) {
            scanf("%d", &allocation[i][j]);
            need[i][j] = maximum[i][j] - allocation[i][j];
}
}
// Calculate remaining resources needed by each account
for (i = 0; i < num_accounts; i++) {
    for (j = 0; j < num_resources; j++) {
        need[i][j] = maximum[i][j] - allocation[i][j];
    }
}
// Check if car loan can be processed
for (i = 0; i < num_accounts; i++) {
    if (is_safe(i)) {
        printf("Account %d can be processed.\n", i);
        return 0;
    }
}
printf("Car loan cannot be processed.\n");
return 0;
}
```

Output:

```
saqib@saqib-VirtualBox:~/Final Paper$ ./3
Enter number of accounts: 3
Enter number of resources: 4
Enter available resources:
1000
2000
3000
4000
Enter maximum resources for each account:
Account 0:
500
1000
2000
3000
Account 1:
300
400
500
600
Account 2:
1000
2000
3000
4000
Enter resources currently allocated to each account:
Account 0:
200
500
1000
1500
Account 1:
100
200
250
300
```


Account 2:
1000
2000
3000
4000
Enter resources currently allocated to each account:
Account 0:
200
500
1000
1500
Account 1:
100
200
250
300
Account 2:
800
1000
2000
2500
Account 0 can be processed.