**C Program: Banker's Algorithm**

**$touch bankers.c**

#include <stdio.h>

#define MAX\_PROCESSES 10

#define MAX\_RESOURCES 10

int allot[MAX\_PROCESSES][MAX\_RESOURCES];

int max[MAX\_PROCESSES][MAX\_RESOURCES];

int need[MAX\_PROCESSES][MAX\_RESOURCES];

int available[MAX\_RESOURCES];

int m, n;

void calculateNeed() {

for (int i = 0; i < m; i++)

for (int j = 0; j < n; j++)

need[i][j] = max[i][j] - allot[i][j];

}

void displayNeed() {

printf("Need Matrix:\n");

for (int i = 0; i < m; i++) {

for (int j = 0; j < n; j++) {

printf("%d ", need[i][j]);

}

printf("\n");

}

}

int isSafe(int safeSeq[]) {

int work[MAX\_RESOURCES];

int finish[MAX\_PROCESSES] = {0};

int count = 0;

for (int i = 0; i < n; i++) {

work[i] = available[i];

}

while (count < m) {

int found = 0;

for (int p = 0; p < m; p++) {

if (!finish[p]) {

int j;

for (j = 0; j < n; j++)

if (need[p][j] > work[j])

break;

if (j == n) {

for (int k = 0; k < n; k++)

work[k] += allot[p][k];

safeSeq[count++] = p;

finish[p] = 1;

found = 1;

}

}

}

if (!found) break;

}

return (count == m);

}

int main() {

int i, j;

printf("Enter the number of resources: ");

scanf("%d", &n);

printf("Enter the number of processes: ");

scanf("%d", &m);

printf("Enter the available matrix:\n");

for (i = 0; i < n; i++) {

printf("Resource %d: ", i);

scanf("%d", &available[i]);

}

printf("Enter the Allocation Matrix:\n");

for (i = 0; i < m; i++) {

printf("Process P%d: ", i);

for (j = 0; j < n; j++) {

scanf("%d", &allot[i][j]);

}

}

printf("Enter the Maximum Matrix:\n");

for (i = 0; i < m; i++) {

printf("Process P%d: ", i);

for (j = 0; j < n; j++) {

scanf("%d", &max[i][j]);

}

}

calculateNeed();

displayNeed();

int safeSeq[MAX\_PROCESSES];

if (isSafe(safeSeq)) {

printf("System is in a safe state. Safe sequence is: ");

for (i = 0; i < m; i++) {

printf("P%d ", safeSeq[i]);

}

printf("\n");

} else {

printf("System is not in a safe state.\n");

}

return 0;

}

**Output:**

**$ gcc -o bankers bankers.c**

**$ ./bankers**

Enter the number of resources: 2

Enter the number of processes: 2

Enter the available matrix:

Resource 0: 2

Resource 1: 3

Enter the Allocation Matrix:

Process P0: 2

4

Process P1: 4

3

Enter the Maximum Matrix:

Process P0: 2

4

Process P1: 5

6

Need Matrix:

0 0

1 3

System is in a safe state. Safe sequence is: P0 P1