Bankerf.c

```
#include <stdio.h>

int main() {
   int p, r, i, j, flag, avail_r[10], allocated_r[10][20], max_r[10][20], need[10][20], exe[10] = {0};

   printf("Enter no of processes and resources: ");
   scanf("%d %d", &p, &r);

   printf("Available resources:\n");
   for (j = 0; j < r; j++) {
      printf("Enter [%d]: ", j);
      scanf("%d", &avail_r[j]);
   }

   printf("Allocated resources:\n");
   for (i = 0; i < p; i++) {
      for (j = 0; j < r; j++) {
         printf("[%d][%d]: ", i, j);
         scanf("%d", &allocated_r[i][j]);
      }
   }

   printf("Max resources:\n");
   for (i = 0; i < p; i++) {
      for (j = 0; j < r; j++) {
         printf("[%d][%d]: ", i, j);
         scanf("%d", &max_r[i][j]);
      }
   }

   printf("Need matrix:\n");
   for (i = 0; i < p; i++) {
      for (j = 0; j < r; j++) {
         need[i][j] = max_r[i][j] - allocated_r[i][j];
         printf("%d\t", need[i][j]);
      }
      printf("\n");
   }

   while (1) {
      for (i = 0; i < p; i++) {
         if (!exe[i]) {
            flag = 1;
            for (j = 0; j < r; j++) {
               if (avail_r[j] < need[i][j]) { flag = 0; break; }
            }
            if (flag) {
               printf("\n%d is running\n", i);
               exe[i] = 1;
```

```
                for (j = 0; j < r; j++) avail_r[j] += allocated_r[i][j];
                break;
            }
        }
    }
    if (i == p) {
        for (flag = 1, i = 0; i < p; i++) if (!exe[i]) flag = 0;
        printf(flag ? "Safe state" : "Not safe");
        break;
    }
  }
  return 0;
}
```