Banker’s Program.

```
#include <stdio.h>

int main() { int numProcesses = 5; // Number of
  processes int numResources = 3; // Number
  of resources

  int allocationMatrix[5][3] = {{1, 1, 2}, {2, 1, 2}, {4, 0, 1}, {0, 2, 0}, {1, 1, 2}}; //
Allocation Matrix int maxMatrix[5][3] = {{4, 3, 3}, {3, 2, 2}, {9, 0, 2}, {7, 5, 3}, {1, 1, 2}};
      // MAX Matrix int availableResources[3] = {2, 1, 0}; // Available Resources

  int isFinished[numProcesses], safeSequence[numProcesses], index = 0;
  for (int k = 0; k < numProcesses; k++) { isFinished[k] = 0;
  }

  int needMatrix[numProcesses][numResources]; for (int i = 0; i
  < numProcesses; i++) { for (int j = 0; j < numResources; j++)
  needMatrix[i][j] = maxMatrix[i][j] - allocationMatrix[i][j];
  }

  for (int k = 0; k < numProcesses; k++) { for (int i = 0; i
    < numProcesses; i++) { if (isFinished[i] == 0) { int
    flag = 0; for (int j = 0; j < numResources; j++) { if
    (needMatrix[i][j] > availableResources[j]) { flag = 1;
    break;
          } } if (flag == 0) {
        safeSequence[index++] = i; for (int y =
        0; y < numResources; y++)
```

```
                        availableResources[y] += allocationMatrix[i][y];
                    isFinished[i] = 1;
                }
            }
        }
    }

    int flag = 1; for (int i = 0; i <
    numProcesses; i++) { if (isFinished[i]
    == 0) { flag = 0; printf("The system is
    not safe.\n"); break;
        }
    }

    if (flag == 1) { printf("SAFE Sequence: "); for (int i
        = 0; i < numProcesses - 1; i++) printf("P%d -> ",
        safeSequence[i]); printf("P%d\n",
        safeSequence[numProcesses - 1]);
    }

    return 0;
}
```

Modified as Taking input from user:

```
#include<stdio.h>

int main(){
   int numofprocess, numofresources;

   printf("Enter the number of processes: ");
   scanf("%d", &numofprocess);

   printf("Enter the number of resources: ");
   scanf("%d", &numofresources);

   int allocationMatrix[numofprocess][numofresources];
   int maxMatrix[numofprocess][numofresources];
   int availableResources[numofresources];

   printf("Enter the allocation matrix:\n");
   for(int i=0; i<numofprocess; i++){
      for(int j=0; j<numofresources; j++){
         scanf("%d", &allocationMatrix[i][j]);
      }
   }

   printf("Enter the max matrix:\n");
   for(int i=0; i<numofprocess; i++){
      for(int j=0; j<numofresources; j++){
         scanf("%d", &maxMatrix[i][j]);
      }
   }

   printf("Enter the available resources:\n");
   for(int i=0; i<numofresources; i++){
      scanf("%d", &availableResources[i]);
   }

   int isFinished[numofprocess], safeSequence[numofprocess], index=0;
   for(int k=0; k<numofprocess; k++){
      isFinished[k]=0;
   }

   int needMatrix[numofprocess][numofresources];
   for(int i=0; i<numofprocess; i++){
      for(int j=0; j<numofresources; j++){
         needMatrix[i][j]=maxMatrix[i][j]-allocationMatrix[i][j];
      }
   }

   for(int k=0; k<numofprocess; k++){
        for(int i=0; i<numofprocess; i++){
            if(isFinished[i]==0){
                int flag = 0;
                for(int j=0; j<numofresources; j++){
                    if(needMatrix[i][j]>availableResources[j]){
                        flag =1;
                        break;
                    }
                }
                if(flag==0){
                    safeSequence[index++]=i;
                    for(int y=0; y<numofresources; y++)
                        availableResources[y]+=allocationMatrix[i][y];
                    isFinished[i]=1;
                }
            }
        }
    }

    int flag =1;
    for(int i=0; i<numofprocess; i++){
        if(isFinished[i]==0){
            flag=0;
            printf("The system is not safe.\n");
            break;
        }
    }
    if(flag==1){
        printf("SAFE Sequence:");
        for(int i=0; i<numofprocess; i++)
            printf("P%d->",safeSequence[i]);
        printf("P%d\n",safeSequence[numofprocess-1]);
    }

    return 0;
}
```