**Assignment 5**

**Name** : Komal Potdar

**Roll No**.: 92

**PRN No**.: 12320165

**Div**: CS B SY

**Batch**: 3

**Banker’s algorithm**

|  |
| --- |
| #include <stdio.h>  int main() {  int n, m, i, j, k;  printf("Enter the number of processes: ");  scanf("%d", &n);  printf("Enter the number of resources: ");  scanf("%d", &m);  int alloc[n][m], max[n][m], avail[m];  printf("Enter the Allocation Matrix:\n");  for (i = 0; i < n; i++) {  for (j = 0; j < m; j++) {  scanf("%d", &alloc[i][j]);  }  }  printf("Enter the Max Matrix:\n");  for (i = 0; i < n; i++) {  for (j = 0; j < m; j++) {  scanf("%d", &max[i][j]);  }  }  printf("Enter the Available Resources:\n");  for (j = 0; j < m; j++) {  scanf("%d", &avail[j]);  }  int f[n], ans[n], ind = 0;  for (k = 0; k < n; k++) {  f[k] = 0;  }  int need[n][m];  for (i = 0; i < n; i++) {  for (j = 0; j < m; j++)  need[i][j] = max[i][j] - alloc[i][j];  }  int y = 0;  for (k = 0; k < n; k++) {  for (i = 0; i < n; i++) {  if (f[i] == 0) {  int flag = 0;  for (j = 0; j < m; j++) {  if (need[i][j] > avail[j]) {  flag = 1;  break;  }  }  if (flag == 0) {  ans[ind++] = i;  for (y = 0; y < m; y++)  avail[y] += alloc[i][y];  f[i] = 1;  }  }  }  }  int flag = 1;  for (int i = 0; i < n; i++) {  if (f[i] == 0) {  flag = 0;  printf("The following system is not safe\n");  break;  }  }  if (flag == 1) {  printf("Following is the SAFE Sequence\n");  for (i = 0; i < n - 1; i++)  printf(" P%d ->", ans[i]);  printf(" P%d", ans[n - 1]);  }  return 0;  } |

**Output:**

