|  |  |  |  |
| --- | --- | --- | --- |
| **Lab No** | 09 | **Reg. No** | 224921 |
| **Student Name** | Muhammad Saad Tariq | **Section** | A |

|  |
| --- |
| Solution Task |
| Code:  //Deadlock avoidance through Banker’s Algorithm  #include <stdio.h>  int main(){  int proc, res, col\_check = 0;  printf("\nEnter no. of processes:");  scanf("%d", &proc);  printf("Enter no. of resources: ");  scanf("%d", &res);  int alloc[proc][res], max[proc][res], need[proc][res], avail[res], seq[proc];  int i,j,a;  for(a = 0; a < proc; a++){  seq[a] = 100;  }  printf("\nAllocate Matrix:\n");  for(i = 0; i < proc; i++){  for(j = 0; j < res; j++){  printf("Allocate[%d][%d]:", i,j);  scanf("%d", &alloc[i][j]);  }  }  printf("\nYour Allocaction matrix:\n");  for(i = 0; i < proc; i++){  for(j = 0; j < res; j++){  printf("%d\t", alloc[i][j]);  }  printf("\n");  }  printf("\nMax matrix:\n");  for(i = 0; i < proc; i++){  for(j = 0; j < res; j++){  printf("Allocate[%d][%d]:", i,j);  scanf("%d", &max[i][j]);  }  }  printf("\nYour Max matrix:\n");  for(i = 0; i < proc; i++){  for(j = 0; j < res; j++){  printf("%d\t", max[i][j]);  }  printf("\n");  }  printf("\nDefine available matrix: ");  for (a = 0; a < res; ++a)  {  scanf("%d", &avail[a]);  }  printf("\nYour available matrix:\n");  for (a = 0; a < res; ++a)  {  printf("%d\t", avail[a]);  }  printf("\nNeed matrix: \n");  for(i = 0; i < proc; i++){  for(j = 0; j < res; j++){  need[i][j] = max[i][j] - alloc[i][j];  }  }  printf("\nYour Need matrix:\n");  for(i = 0; i < proc; i++){  for(j = 0; j < res; j++){  printf("%d\t", need[i][j]);  }  printf("\n");  }  printf("\nExecution order:\n");  int count = 0;  int check = 0;  //alotting  while(count < proc){  for(i = 0; i < proc; i++){  for(j = 0; j < res; j++){  if(need[i][j] <= avail[j]){  col\_check++;  }  }  if(col\_check == res){  for(a = 0; a < proc; a++){  if(seq[a] == i)  goto label;  }  seq[count] = i;  count++;  if(count > 5)  break;  printf("Process[%d] executed\n", i);  col\_check = 0;  printf("Process[%d] finished\n", i);  for(j = 0; j < res; j++){  avail[j] += alloc[i][j];  }  printf("\nAvailable matrix:\n");  for(a = 0; a < res; a++)  printf("%d\t", avail[a]);  printf("\n");  }  else{  label:  col\_check = 0;  }  }  for(a = 0; a < proc; a++){  if(seq[a] == 100)  check++;  }  if(check == proc){  printf("\n\nDeadlock has occured!\n");  exit(0);  }  }  printf("\n\nDeadlock Prevented! (By deadlock avoidance)\n");  return 0;  }  Output:  Deadlock Avoidance:      Deadlock Occurrence: |