Q.24) #include <stdio.h>

Int main()

{

Int x,y,z,k,I;

N= 5 ;// no of processes

M= 4 ;// no of resources

Int alloc[5][4 ]={ { 0,0,1,2 } //Allocation matrix

{ 1,0,0,0 }

{ 1,3,5,4 }

{ 0,6,3,2 }

{ 0,0,1,4 } }

Int max[5][4]=]={ { 0,0,1,2 } //Max matrix

{1,7,5,0 }

{ 2,3,5,6 }

{ 0,6,5,2 }

{ 0,6,5,6 } }

Int avail[] = { 1,5,2,0 };Available resource

Int f[x] ,ans[x],ind=0;

For(k=0; k<n; k++ ) {

F[k] =0;

}

Int need[x][y];

For (i=0 ;i<x ; i++) {

For(z=0 ;z <y ; j++)

Need[i][z] = max[i][z]-alloc [i][z];

}

Int p=0;

For (k=0 ;k< 5 ; k++) {

For(i=0; i<x ; i++) {

If( f[i] == 0 ) {

Int flag =0;

For (z=0;z< y ; j++) {

If(need [i][z] >avail[z] ){

Flag = 1;

Break;

}

}

If (flag ==0) {

And[ind ++] = I;

For (p=0; p< y; p++)

Avail[p] +=alloc[i][p];

F[i] = 1;

}

}

}

}

Print(“ given question in safe sequence\n”);

For(i=0; i<x-1 ;i++)

Printf(“ p%d ->” ,ans[i]);

Printf(“ p%d “ ,ans[x-1] ) ;

Return (0) ;

Q.3)#include <stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<pthread.h>

#define m 30

#define MAX\_THREADS 6

Int prime\_arr[m]={0};

Void \*printprime(void \*ptr)

{

Int a, flag;

Int b=(int)(long long int)ptr;

for(b=2 ;b<m ; b++){

flag=0;

for(a=2; a<=b/2 ; j++)

{

If(b%a==0)

{flag=1;

Break;

}

}

If(flag==0)

{

Prime\_arr[b]=1;

}

}

}

Int main()

{

Pthread\_t tid[max\_threads]={0};

Int count =0;

For(count=0; count<max-threads; count++)

{

Printf(\n creating threads %d , count);

Pthread\_create(&tid[count],null ,printprime ,(void\*)count);

}

Printf(“\n”)

For(count =0; count<max\_threads;count++)

{

Pthread\_join(tid[count],null);

}

Int k=0

For(count=0 ;count<m; count++)

If (prime\_arr[count]==1)

Print(“%d “, count);

Return 0;

}