**Resource Request Algorithm**

#include <stdio.h>

int main()

{

// P0, P1, P2, P3, P4 are the Process names here

int n, m, i, j, k, alloc[10][10], max[10][10],avail[10],req[10],u,r,request;

printf("Enter number of processes:\n");

scanf("%d",&n);

printf("Enter number of resources:\n");

scanf("%d",&m);

printf("Enter the allocation values for each resource in the %d processes:\n", n);

for(u=0;u<n;u++)

{

for(r=0;r<m;r++)

{

scanf("%d",&alloc[u][r]);

}

}

printf("Enter the maximum values for each resource in the %d processes:\n", n);

for(int u=0;u<n;u++)

{

for(int r=0;r<m;r++)

{

scanf("%d",&max[u][r]);

}

}

printf("Enter the available values for each resource:\n", n);

for(int r=0;r<m;r++)

{

scanf("%d",&avail[r]);

}

printf("Enter the number of the process for which you want to request:\n", n);

scanf("%d",&request);

printf("Enter the request values for each resource:\n", n);

for(int r=0;r<m;r++)

{

scanf("%d",&req[r]);

if (req[r] > avail[r]){

printf("Request is greater than available");

return 0;

}

else {avail[r]-=req[r];

alloc[request][r]+=req[r];}

}

int f[n], ans[n], ind = 0;

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

f[k] = 0;

}

printf("The Need matrix:\n");

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];

printf("%d ",need[i][j]);}

printf("\n");

}

int y = 0;

for (k = 0; k < 5; 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;

}

}

}

}

printf("Following is the SAFE Sequence\n");

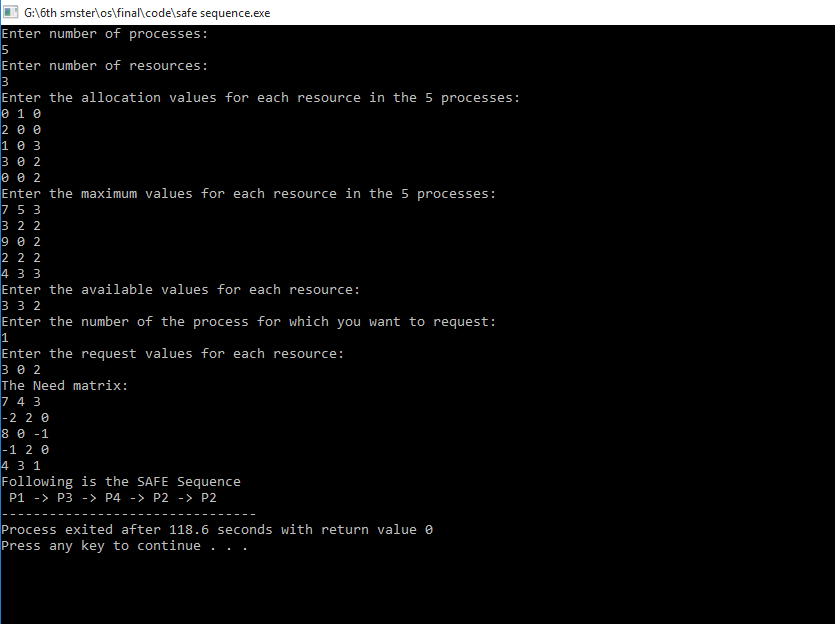
for (i = 0; i < n - 1; i++)

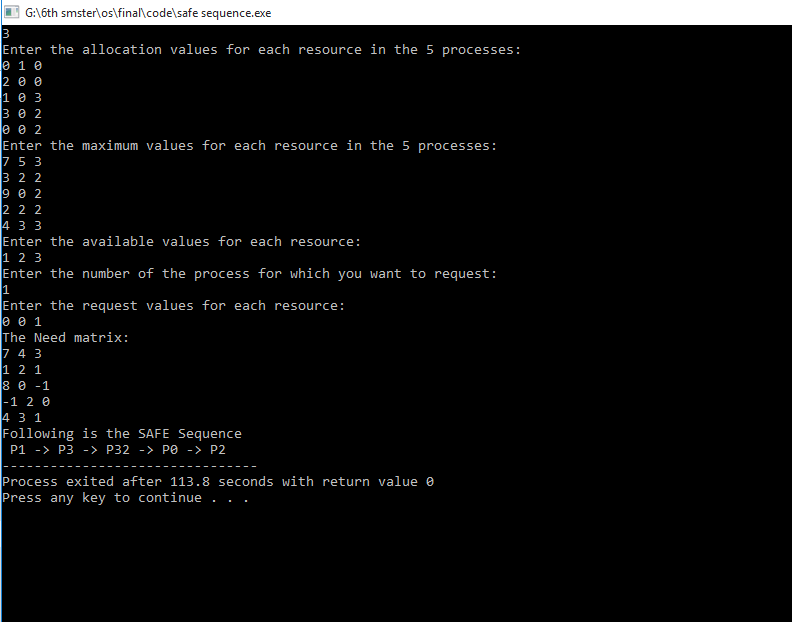
printf(" P%d ->", ans[i]);

printf(" P%d", ans[n - 1]);

return (0);

}

**Output:**

****