#include <stdio.h>

#include <stdlib.h>

#define m 50

int no1;

int no2;

int allocation[m][m], available[m], max[m][m], need[m][m];

int r;

int main()

{

void check();

void print();

int i, j, p = 0, q = 0;

char ch;

int sangle;

int request[m], allocation1[m][m], need1[m][m], available1[m];

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\* the design and implementation of banker algorithm \*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("please input the total number of process\n");

scanf("%d", &no1);

printf("please input the total number of resources\n");

scanf("%d", &no2);

printf("please input max matrix:\n");

for (i = 0; i < no1; i++)

for (j = 0; j < no2; j++)

scanf("%d", &max[i][j]);

printf("please input allocation matrix：\n");

for (i = 0; i < no1; i++)

for (j = 0; j < no2; j++)

scanf("%d", &allocation[i][j]);

for (i = 0; i < no1; i++)

for (j = 0; j < no2; j++)

need[i][j] = max[i][j] - allocation[i][j];

printf("please input available matrix\n");

for (i = 0; i < no2; i++)

scanf("%d", &available[i]);

print();

check();

if (r == 1)

{

do

{

q = 0;

p = 0;

printf("\n please input the process id of request resource(0~4):\n");

for (j = 0; j <= 10; j++)

{

scanf("%d", &i);

if (i >= no1)

{

printf("input error,please input again:\n");

continue;

}

else

{

break;

}

}

printf("\n please input the requested resource of the process request[j]:\n");

for (j = 0; j < no2; j++)

scanf("%d", &request[j]);

for (j = 0; j < no2; j++)

if (request[j] > need[i][j])

p = 1;

if (p)

printf("request failure！\n");

else

{

for (j = 0; j < no2; j++)

if (request[j] > available[j])

q = 1;

if (q)

printf("request flilure！\n");

else

{

for (j = 0; j < no2; j++)

{

available1[j] = available[j];

allocation1[i][j] = allocation[i][j];

need1[i][j] = need[i][j];

available[j] = available[j] - request[j];

allocation[i][j] += request[j];

need[i][j] = need[i][j] - request[j];

}

print();

check();

if (r == 0)

{

for (j = 0; j < no2; j++)

{

available[j] = available1[j];

allocation[i][j] = allocation1[i][j];

need[i][j] = need1[i][j];

}

printf("return resources before allocation\n");

print();

}

}

}

printf("\n need continue allocation？y or n ?\n");

scanf("%c\n", &ch);

printf("\n");

} while (ch == 'y' || ch == 'Y');

}

return 0;

}

void check()

{

int k, f, v = 0, i, j;

int work[m], a[m];

int finish[m];

r = 1;

for (i = 0; i < no1; i++)

finish[i] = 0;

for (i = 0; i < no2; i++)

work[i] = available[i];

k = no1;

do

{

for (i = 0; i < no1; i++)

{

if (finish[i] == 0)

{

f = 1;

for (j = 0; j < no2; j++)

if (need[i][j] > work[j])

f = 0;

if (f == 1)

{

finish[i] = 1;

a[v++] = i;

for (j = 0; j < no2; j++)

work[j] += allocation[i][j];

}

}

}

k--;

} while (k > 0);

f = 1;

for (i = 0; i < no1; i++)

{

if (finish[i] == 0；

{

f = 0;

break;

}

}

if (f == 0)

{

printf("system is not safe!\n");

r = 0;

}

else

{

printf("\n system is safe,the safe squen:\n");

for (i = 0; i < no1; i++)

printf("p%d", a[i]);

}

}

void print()

{

int i, j;

printf("\n");

printf("\*\*\*\*\*\*\*\*\*the resource allocation is\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("process id | max |allocation | need |\n");

for (i = 0; i < no1; i++)

{

printf(" p%d%d ", i, i);

for (j = 0; j < no2; j++)

{

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

}

for (j = 0; j < no2; j++)

{

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

}

for (j = 0; j < no2; j++)

{

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

}

printf("\n");

}

printf("\n");

printf("the available resources:");

for (j = 0; j < no2; j++)

{

printf(" %d", available[j]);

}

printf("\n");

}