PROJECT JS

MAROOM ALREFAI

0205262

```
#include <iostream>

using namespace std;


int main() {

   cout<<"this program for calculate job sequance of two or three machines and at the maximum ten jobs"<<endl;

   cout<<"please enter the number of jobs"<<endl;

    int x ; //number of job

    cin>>x;

    if (x>10 || x<2)

    cout <<"you can enter less than ten jobs and up to two only"<<endl;

    else{

       int y=0;

       cout<<"please enter the number of machines"<<endl;

       cin>>y;

       if (y != 2 && y != 3)

       cout<<"you can enter only two machines or three machines"<<endl;

       else

       if (y==3)

       {

   int A[10] = {0}; //machine A

   int B[10] = {0}; //machine B

   int C[10] = {0}; //machine C

   int A1[10] = {0};
```

```
int B1[10] = {0};
int C1[10] = {0};
cout<<"please enter the jobs of machine A"<<endl;
 for (int i = 0; i < x; i++)
 {
    cin>>A[i];
    A1[i]=A[i];
 }
 cout<<"please enter the jobs of machine B"<<endl;
 for (int i = 0; i < x; i++)
 {
    cin>>B[i];
     B1[i]=B[i];
 }
 cout<<"please enter the jobs of machine C"<<endl;
 for (int i = 0; i < x; i++)
 {
    cin>>C[i];
      C1[i]=C[i];
 }



int optimal[] = {0}; //optimal array


int solution[x+1][11]= {0}; //two dimentional array for solution
int AB[x] ={0} , BC[x]={0};


int right = x-1;
int left = 0;
```

```
   int minIndex = -1;
   char N = 'f';

   for(int i=0;i<x;i++) //calculate AB and BC
   {
      AB[i]=A[i]+B[i];
      BC[i]=B[i]+C[i];
   }
   int job =x;

while(job) //find the minmum
{
    int min = 10e3;
   for (int i = 0; i < x; i++)
   {

      if (AB[i] < min && AB[i] > 0) {
         min = AB[i];
         minIndex = i;
         N = 'a';
      }

      if (BC[i] < min && BC[i] > 0) {
         min = BC[i];
         minIndex = i;
         N = 'b';
      }

      if (BC[i] == min && BC[i] > 0 && AB[i] > 0&& AB[i]<AB[minIndex])      //here if jobs are equal
```

```
    {
      min = BC[i];
      minIndex = i;
      N = 'b';
    }
   else{
       if (AB[i] == min && AB[i] > 0 && BC[i] > 0 && BC[i]<BC[minIndex])
    {
        min = AB[i];
      minIndex = i;
      N = 'a';
   }

    }

}

AB[minIndex] = 0;      //delete if I found the minmum
        BC[minIndex] = 0;

  if (N =='a') {  //fell the optimal array
     optimal[left] = minIndex+1;
     left++;
     N = 'f';
  }

  if (N == 'b') {
     optimal[right] = minIndex+1;
     right--;
```

```
          N = 'f';
        }
    job--;
}



    cout << "Optimal array: ";  //print the optimal
    for (int i = 0; i < x; i++) {
        cout << optimal[i] << " ";
    }


    cout << endl;
    cout << endl;
    cout<<"job"<<"\t Machine A"<<"\t Machine B"<<"\t Machine C"<<"\t idlE time"<<endl;  //print the
header
    cout<<"\t in"<<"\tout"<<"\t in"<<"\tout"<<"\t in"<<"\tout"<<"\t A"<<"\t B"<<"\t C"<<endl;




    for(int i=0;i<x;i++){
        int j=0;
       if(j<10){
        int job=optimal[i];
        solution[i][j] = job;  //optimal order for job (column=1)
        if(i==0)
        solution[i][j+1] = 0; //first in of first job =0 (column=2)
        else
        solution[i][j+1] = solution[i-1][j+2]; // in column of A machine (column=2)
```

```
solution[i][j+2] = solution[i][j+1]+A1[job-1];// out column of A machine (column=3)



//in column of B machine (column=4)
if(i==0)
{
   solution[i][j+3] = solution[i][j+2];
   if(solution[i][j+3]>0)                    //if in time not equal 0
   solution[i][j+8] =solution[i][j+3];
}
else if(solution[i-1][j+4] > solution[i][j+2])
{
   solution[i][j+3] = solution[i-1][j+4];
}
else
{
   solution[i][j+3] = solution[i][j+2];
   solution[i][j+8] = solution[i][j+2] - solution[i-1][j+4]; //idel time of machine B
}

//out column of machine B (column=5)
solution[i][j+4] = solution[i][j+3]+ B1[job-1];




//in column of C machine (column=6)
if(i==0)
{
   solution[i][j+5] = solution[i][j+4];
```

```
    if(solution[i][j+4]>0)                  //if in time not equal 0
    solution[i][j+9] = solution[i][j+5];
  }

  else if(solution[i-1][j+6] < solution[i][j+4])
  {
    solution[i][j+5] = solution[i][j+4];
    solution[i][j+9] = solution[i][j+4] - solution[i-1][j+6]; //idel time of machine C
  }
  else
    solution[i][j+5] = solution[i-1][j+6];



  //out column of machine C (column=7)
  solution[i][j+6] = solution[i][j+5]+ C1[job-1];

  if(i==4) // idel time of machine A
  solution[i][j+7] = solution[i][j+6] - solution[i][j+2];
  j++;
}}

for(int i=0;i<x;i++){   //print solution
  for(int j=0;j<10;j++){
  cout<<solution[i][j]<<"\t"<<" ";
}
cout<<endl;
}
```

```
int total_B=0;
int total_C=0;
for(int i=0;i<x;i++){   //total idle for B machine
   int j=8;
   int k=9;
   total_B+=solution[i][j];
   total_C+=solution[i][k];
}
cout<<endl;
cout<<endl;
cout<<"Total elaspe time:"<<solution[x-1][6]<<endl;
cout<<"elaspe time: A's idle time: "<<solution[x-1][6]-solution[x-1][2]<<endl;
cout<<"elaspe time: B's idle time: "<<total_B + (solution[x-1][6]-solution[x-1][4])<<endl;
cout<<"elaspe time: c's idle time: "<<total_C + (solution[x-1][6]-solution[x-1][6])<<endl;

cout<<endl;
    }

////////////////////////////////////////////////////////////////////////////////////////////////////

    if(y==2)
    {
int A[x] = {0}; //machine A
int B[x] = {0}; //machine B
int A1[x] = {0};
int B1[x] = {0};

cout<<"please enter the jobs of machine A"<<endl;
 for (int i = 0; i < x; i++)
```

```
    {
       cin>>A[i];
       A1[i]=A[i];
    }
    cout<<"please enter the jobs of machine B"<<endl;
    for (int i = 0; i < x; i++)
    {
       cin>>B[i];
        B1[i]=B[i];
    }

   int optimal[] = {0}; //optimal array

   int solution[x+1][7]= {0}; //two dimentional array for solution

   int right = x-1;
   int left = 0;
   int minIndex = -1;
   char N = 'f';
   int job =x;
   int x1=x;

while(job) //find the minmum
{
    int min = 10e3;
   for (int i = 0; i < x1; i++)
   {

      if (A[i] < min && A[i] > 0) {
```

```
        min = A[i];
        minIndex = i;
        N = 'a';
    }

    if (B[i] < min && B[i] > 0) {
        min = B[i];
        minIndex = i;
        N = 'b';
    }

    if (B[i] == min && B[i] > 0 && A[i] > 0&& A[i]<A[minIndex])      //here if jobs are equal
     {
        min = B[i];
        minIndex = i;
        N = 'b';
     }
    else{
        if (A[i] == min && A[i] > 0 && B[i] > 0 && B[i]<B[minIndex])
     {
         min = A[i];
        minIndex = i;
        N = 'a';
     }

     }

}
```

```
    A[minIndex] = 0;       //delete if I found the minmum
            B[minIndex] = 0;

       if (N =='a') {  //fell the optimal array
          optimal[left] = minIndex+1;
          left++;
          N = 'f';
       }

       if (N == 'b') {
          optimal[right] = minIndex+1;
          right--;
          N = 'f';
       }
    job--;
}


    cout << "Optimal array: ";  //print the optimal
    for (int i = 0; i < x; i++) {
       cout << optimal[i] << " ";
    }

    cout << endl;
    cout << endl;
    cout<<"job"<<"\t Machine A"<<"\t Machine B"<<"\t idlE time"<<endl;  //print the header
    cout<<"\t in"<<"\tout"<<"\t in"<<"\tout"<<"\t A"<<"\t B"<<endl;

    for(int i=0;i<x;i++){
```

```
 int j=0;
if(j<10){
 int job2=optimal[i];
 solution[i][j] = job2;  //optimal order for job (column=1)
 if(i==0)
 solution[i][j+1] = 0; //first in of first job =0 (column=2)
 else
 solution[i][j+1] = solution[i-1][j+2]; // in column of A machine (column=2)

 solution[i][j+2] = solution[i][j+1]+A1[job2-1];// out column of A machine (column=3)


//in column of B machine (column=4)
 if(i==0)
 {
    solution[i][j+3] = solution[i][j+2];
    if(solution[i][j+3]>0)                  //if in time not equal 0
    solution[i][j+6] =solution[i][j+3];
 }
 else if(solution[i-1][j+4] > solution[i][j+2])
 {
    solution[i][j+3] = solution[i-1][j+4];
 }
 else
 {
    solution[i][j+3] = solution[i][j+2];
    solution[i-1][j+6] = solution[i][j+2] - solution[i-1][j+4]; //idel time of machine B
 }
```

```
   //out column of machine B (column=5)
   solution[i][j+4] = solution[i][j+3]+ B1[job2-1];

   if(i==x-1) // idel time of machine A
   solution[i][j+5] = solution[i][j+4] - solution[i][j+2];
   j++;
}}

for(int i=0;i<x;i++){   //print solution
   for(int j=0;j<7;j++){
   cout<<solution[i][j]<<"\t"<<" ";
}
cout<<endl;
}

int total_B=0;
for(int i=0;i<x;i++){   //total idle for B machine
   int j=6;
total_B+=solution[i][j];
}
cout<<endl;
cout<<endl;
cout<<"Total elaspe time: "<<solution[x-1][4]<<endl;
cout<<"elaspe time: A's idle time: "<<solution[x-1][5]<<endl;
cout<<"elaspe time: B's idle time: "<<total_B<<endl;
cout<<endl;
    }
 }
return 0;
```

}