ONE

#include<bits/stdc++.h>

using namespace std;

void firstFit(int blocks[], int n,int processes[], int m){

int allocation[m];

for(int i=0;i<m;i++){

allocation[i]=-1;

}

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

{

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

{

if (blocks[j] >= processes[i])

{

allocation[i] = j;

blocks[j] -= processes[i];

break;

}

}

}

for(int i=0;i<m;i++){

if(allocation[i]!=-1){

cout<<"Process "<<i+1<<" -> "<<allocation[i]+1<<endl;

}else{

cout<<"Process "<<i+1<<" -> "<<"Unallocated"<<endl;

}

}

}

int main()

{

int n,m;

cout<<"Sarthak bhatia\n07714803118"<<endl;

cout<<"Enter the number of blocks: ";

cin>>n;

cout<<"\nEnter the number of processes: ";

cin>>m;

int blocks[n];

int processes[m];

cout<<"Enter block sizes: "<<endl;

for(int i=0;i<n;i++){

cin>>blocks[i];

}

cout<<"Enter processes sizes: "<<endl;

for(int i=0;i<m;i++){

cin>>processes[i];

}

firstFit(blocks, n, processes, m);

return 0 ;

}

TWO

#include<bits/stdc++.h>

using namespace std;

void bestFit(int blocks[], int n,int processes[], int m){

int allocation[m];

for(int i=0;i<m;i++){

allocation[i]=-1;

}

for (int i=0; i<m; i++){

int Idx = -1;

for (int j=0; j<n; j++){

if (blocks[j] >= processes[i])

{

if (Idx == -1)

Idx = j;

else if (blocks[Idx] > blocks[j])

Idx = j;

}

}

if (Idx != -1){

allocation[i] = Idx;

blocks[Idx] -= processes[i];

}

}

for(int i=0;i<m;i++){

if(allocation[i]!=-1){

cout<<"Process "<<i+1<<" -> "<<allocation[i]+1<<endl;

}else{

cout<<"Process "<<i+1<<" -> "<<"Unallocated"<<endl;

}

}

}

int main()

{

int n,m;

cout<<"---------Best Fit---------"<<endl;

cout<<"Sarthak bhatia\n07714803118"<<endl;

cout<<"Enter the number of blocks: ";

cin>>n;

cout<<"\nEnter the number of processes: ";

cin>>m;

int blocks[n];

int processes[m];

cout<<"Enter block sizes: "<<endl;

for(int i=0;i<n;i++){

cin>>blocks[i];

}

cout<<"Enter processes sizes: "<<endl;

for(int i=0;i<m;i++){

cin>>processes[i];

}

bestFit(blocks, n, processes, m);

return 0 ;

}

THREE

#include<bits/stdc++.h>

using namespace std;

void worstFit(int blocks[], int n,int processes[], int m){

int allocation[m];

for(int i=0;i<m;i++){

allocation[i]=-1;

}

for (int i=0; i<m; i++){

int Idx = -1;

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

{

if (blocks[j] >= processes[i])

{

if (Idx == -1)

Idx = j;

else if (blocks[Idx] < blocks[j])

Idx = j;

}

}

if (Idx != -1){

allocation[i] = Idx;

blocks[Idx] -= processes[i];

}

}

for(int i=0;i<m;i++){

if(allocation[i]!=-1){

cout<<"Process "<<i+1<<" -> "<<allocation[i]+1<<endl;

}else{

cout<<"Process "<<i+1<<" -> "<<"Unallocated"<<endl;

}

}

}

int main()

{

int n,m;

cout<<"---------Worst Fit---------"<<endl;

cout<<"Sarthak bhatia\n07714803118"<<endl;

cout<<"Enter the number of blocks: ";

cin>>n;

cout<<"\nEnter the number of processes: ";

cin>>m;

int blocks[n];

int processes[m];

cout<<"Enter block sizes: "<<endl;

for(int i=0;i<n;i++){

cin>>blocks[i];

}

cout<<"Enter processes sizes: "<<endl;

for(int i=0;i<m;i++){

cin>>processes[i];

}

worstFit(blocks, n, processes, m);

return 0 ;

}