Pre-emptive

#include <bits/stdc++.h>

using namespace std;

struct Process

{

int process\_no;

int burst\_time;

int arrival\_time;

};

void Find\_Waiting\_Time(Process proc[], int n, int wt[])

{

int rt[n];

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

rt[i] = proc[i].burst\_time;

int complete = 0, t = 0, minm = INT\_MAX;

int shortest = 0, finish\_time;

bool cases = false;

while (complete != n)

{

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

{

if ((proc[j].arrival\_time <= t) &&

(rt[j] < minm) && rt[j] > 0)

{

minm = rt[j];

shortest = j;

cases= true;

}

}

if (cases== false)

{

t++;

continue;

}

rt[shortest]--;

minm = rt[shortest];

if (minm == 0)

minm = INT\_MAX;

if (rt[shortest] == 0)

{

complete++;

cases = false;

finish\_time = t + 1;

wt[shortest] = finish\_time - proc[shortest].burst\_time -proc[shortest].arrival\_time;

if (wt[shortest] < 0)

wt[shortest] = 0;

}

t++;

}

}

void Find\_Average\_Time(Process proc[], int n)

{

int wt[n], tat[n], total = 0,total\_tat = 0;

Find\_Waiting\_Time(proc, n, wt);

cout << "Process No " << " Waiting time " << "\n";

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

{

total = total+ wt[i];

cout << " " << proc[i].process\_no << "\t\t" << wt[i]

<< "\t\t " << endl;

}

cout << "\nThe Average waiting time of the given process is = " << (1.0\*total)/ (1.0\*n) << "\n";

}

int main()

{

int n;

cin >> n;

Process p[n];

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

{

cin >> p[i].arrival\_time;

p[i].process\_no = i + 1;

}

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

cin >> p[i].burst\_time;

}

Find\_Average\_Time(p, n);

return 0;

}