#include<stdio.h>

#include<stdlib.h>

struct process{

int pid;

int burst\_time;

};

void swap(struct process \*xp, struct process \*yp)

{

struct process temp = \*xp;

\*xp = \*yp;

\*yp = temp;

}

void sjf(struct process p[], int n)

{

int i, j;

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

{

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

{

if (p[j].burst\_time > p[j+1].burst\_time)

{

swap(&p[j], &p[j+1]);

}

}

}

printf("\nOrder of execution:\n");

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

printf("P%d ",p[i].pid);

}

float avg\_wt=0, avg\_tat=0;

int wt[n],tat[n];

wt[0]=0;

tat[0]=p[0].burst\_time;

for(i=1;i<n;i++){

wt[i]=wt[i-1]+p[i-1].burst\_time;

tat[i]=tat[i-1]+p[i].burst\_time;

avg\_wt+=wt[i];

avg\_tat+=tat[i];

}

avg\_wt/=n;

avg\_tat/=n;

printf("\n\nAverage waiting time: %0.2f",avg\_wt);

printf("\nAverage turnaround time: %0.2f",avg\_tat);

}

int main(){

int n,i;

printf("Enter the number of processes: ");

scanf("%d",&n);

struct process p[n];

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

printf("\nProcess %d\n",i+1);

p[i].pid=i+1;

printf("Enter burst time: ");

scanf("%d",&p[i].burst\_time);

}

sjf(p,n);

return 0;

}

OUTPUT:

Enter the number of processes: 4

Process 1

Enter burst time: 5

Process 2

Enter burst time: 6

Process 3

Enter burst time: 8

Process 4

Enter burst time: 9

Order of execution:

P1 P2 P3 P4

Average waiting time: 8.75

Average turnaround time: 14.50

--------------------------------

Process exited after 7.167 seconds with return value 0