#include<stdio.h>

struct Process

{

int processId;

int completionTime;

int arrivalTime;

int burstTime;

};

typedef struct Process process;

void swap(process\* i, process\* j)

{

process temp = \*i;

\*i = \*j;

\*j = temp;

}

void bubbleSort(process\* arr, int size)

{

int j,i,swapped;

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

{

swapped =0;

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

{

if(arr[j].arrivalTime > arr[j+1].arrivalTime)

{

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

swapped++;

}

}

if(swapped==0) break;

}

}

int turnAroundTime(int arrivalTime, int completionTime)

{

return(completionTime - arrivalTime);

}

int waitingTime(int arrivalTime, int completionTime, int burstTime)

{

int temp = turnAroundTime(arrivalTime,completionTime);

return(temp-burstTime);

}

int main()

{

int size = 2,i;

process p[size];

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

{

printf("Enter processId, arrivalTime, completionTime, burstTime of process\n");

scanf("%d%d%d%d",&p[i].processId,&p[i].arrivalTime,&p[i].completionTime,&p[i].burstTime);

}

bubbleSort(p,size);

printf("processId\tarrivalTime\tburstTime\tcompletionTime\tturnAroundTime\twaitingTime\n");

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

{

printf("%-10d\t%-10d\t%-10d\t%-10d\t%-10d\t%-10d\n",p[i].processId, p[i].arrivalTime, p[i].burstTime, p[i].completionTime, turnAroundTime(p[i].arrivalTime,p[i].completionTime), waitingTime(p[i].arrivalTime,p[i].completionTime,p[i].burstTime));

}

}