Contents

[**USCS3PO1: USCS303 – Operating System (0S)- Practical -02** 1](#_Toc77967800)

[**Shortest Job first Scheduling Algorithm (SJF) Algorithm** 1](#_Toc77967801)

[**Practical Date: 23rd July ,2021 (Friday)** 1](#_Toc77967802)

[**Practical Aim:** 1](#_Toc77967803)

[**Algorithm** 2](#_Toc77967804)

[**Solved Example** 2](#_Toc77967805)

[**Implementation** 7](#_Toc77967806)

[**Input:** 11](#_Toc77967807)

[**Output:** 12](#_Toc77967808)

[**Sample Output(Screenshots of all the examples)** : 13](#_Toc77967809)

# 

# **USCS3PO1: USCS303 – Operating System (0S)- Practical -02**

## **Shortest Job first Scheduling Algorithm** **(SJF) Algorithm**

### **Practical Date: 23rd July ,2021 (Friday)**

### **Practical Aim:**

Implement SJF (with no preemption ) scheduling algorithm in Java.

**Shortest Job First(SJF**) CPU scheduling algorithm is a CPU scheduling algorithm which is also called **as Shortest Job Next** .

It is Both Non Preemptive and Preemptive CPU Schedulin algorithm.

For Non Preemptive(with no preepmtive) approach, key idea is to allocate the CPU to the process with the smallest burst time so that the CPU seems to be more responsive.

Burst time is the amount of time required by a process for its execution on the CPU .

More appropriate term for this scheduling method would be the shortest next CPU burst algorithm, because scheduling depends on the length of the next CPU burst of a process rather than its total length.

### **Algorithm**

**Step 1:**Input the number of processes required to be scheduled using SJF , burst time for each process .

**Step 2:** Using enhanced bubble sort technique ,sort the all given process in ascending order according to burst time in a ready queue.

**Step 3:**Calculate the Finish Time ,Turn Around Time and Waiting Time for each process which in turn help to calculate Average Waiting Time and Average Turn Around Time required by CPU to schedule given set of process using SJF.

**Step 3.1:** for i= 0, Finish Time T0 = Arrival Time T0 + Burst Time T0

**Step 3.2:** for i >= 1, Finish Time Ti = Arrival Time Ti + Burst Time Ti-1

**Step 3.3:** for i = 0, Turn Around Time T0 =Finish Time T0 – Arrival Time T0

**Step 3.4:** for i > = 1, Turn Around Time Ti =Finish Time Ti – Arrival Time Ti

**Step 3.5:** for i = 0, Waiting Time T0 =Turn Around Time T0 – Burst Time T0

**Step 3.6:** for i >=1, Waiting Time Ti =Turn Around Time Ti – Burst Time Ti-1

**Step 4:** Process with less arrival time comes first and gets schedule first by the CPU.

**Step 5:**Calculate the Average Waiting Time and Average Turn Around Time .

**Step 6:**Stop.

### **Solved Example**

1. Consider following example containing five process arrive at same time.

|  |  |
| --- | --- |
| Process ID | Burst Time |
| P0 | 6 |
| P1 | 3 |
| P2 | 8 |
| P3 | 3 |
| P4 | 4 |

**Step 1:**Processesget executed according to their lowest burst time first.

|  |  |
| --- | --- |
| Process ID | Burst Time |
| P1 | 3 |
| P3 | 3 |
| P4 | 4 |
| P0 | 6 |
| P2 | 8 |

**Step 2:**Following shows the scheduling and execution of processes.

**Step 2.1:**At start P1 has shortest execution time which is 0-3 seconds.

System Time :0

Process Schedule :P1

Finish Time : 0 + 3 =3

Waiting Time :3 - 3 = 0

Turn Around Time :3 – 0 = 3

**Step 2.2 :**Next shortest exection time is process P3 for duration 3-6 seconds.

System Time :3

Process Schedule :P1, P3

Finish Time : 3 + 3 =6

Waiting Time :6 - 3 = 3

Turn Around Time :6 – 0=6

**Step 2.3 :**Next job with shortest execution time is P4 for a Duration 6-10 seconds.

System Time : 6

Process Schedule :P1,P3,P4

Finish Time : 6 + 4 =10

Waiting Time :10 - 4 = 6

Turn Around Time : 10 – 0=10

**Step 2.4 :**Next job with shortest execution time is P0 for duration of 10-16 seconds.

System Time :10

Process Schedule :P1,P3,P4,P0,P2

Finish Time : 16 + 8 =24

Waiting Time :24 – 8 = 16

Turn Around Time :24 – 0 =24

**Step 2.5 :**Similarly, next job with shortest execution time is P2 for duration of 16-24 seconds

**Step 3:** Calculate the Average Waiting Time and Average Turn Around Time.

Average Waiting Time =(0 + 3 + 6 + 10 + 16)/5

=35/5

=7

Average Turn Around Time =(3 + 6 + 10 + 16 + 24)/5

= 59/5

=11.8

**Step 4:**After scheduling of all provided processes:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Process  ID | Burst  Time | Arrival  Time | Finish Time  [Prev. Finish Time + Burst  Time] | Turn Around  Time  [FinishTime-Arrival Time] | Waiting Time  [TurnAround  Time -Burst Time] |
| P1 | 3 | 0 | (-+3=)3 | (3-0=)6 | (6-6=)0 |
| P3 | 3 | 0 | (3+3=)6 | (9-0=)9 | (9-3=)6 |
| P4 | 4 | 0 | (6+4=)10 | (17-0=)17 | (17-8=)9 |
| P0 | 6 | 0 | (10+6=)16 | (20-0=)20 | (20-3=)17 |
| P2 | 8 | 0 | (16+8=)24 | (24-0=)24 | (24-4=)20 |
|  |  |  |  |  |  |
| Average |  |  |  | 11.80000000 | 7.00000000 |

**Gnatt Chart**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **P1** | **P3** | **P4** | **P0** | **P2** |

**0 3 6 10 16 24**

1. Consider following example containing three process arrive at same time

|  |  |
| --- | --- |
| Process  ID | Burst  Time |
| P0 | 2 |
| P1 | 1 |
| P2 | 6 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Process  ID | Burst  Time | Arrival  Time | Finish Time  [Prev. Finish Time + Burst  Time] | Turn Around  Time  [FinishTime-Arrival Time] | Waiting Time  [TurnAround  Time -Burst Time] |
| P1 | 1 | 0 | (-+1=)1 | (1-0=)1 | (1-1=)0 |
| P0 | 2 | 0 | (1+2=)3 | (3-0=)3 | (3-2=)1 |
| P2 | 6 | 0 | (3+6=)9 | (9-0=)9 | (9-6=)3 |
| Average |  |  |  | 4.33333334 | 1.33333334 |

**Gnatt Chart**

|  |  |  |
| --- | --- | --- |
| **P1** | **P0** | **P2** |

**0 1 3 9**

1. Consider the following example containing five processes arrive at same time .

|  |  |
| --- | --- |
| Process ID | Burst Time |
| P0 | 7 |
| P1 | 3 |
| P2 | 2 |
| P3 | 10 |
| P4 | 8 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Process  ID | Burst  Time | Arrival  Time | Finish Time  [Prev. Finish Time + Burst  Time] | Turn Around  Time  [FinishTime-Arrival Time] | Waiting Time  [TurnAround  Time -Burst Time] |
| P2 | 2 | 0 | (-+2=)2 | (2-0=)2 | (2-2=)0 |
| P1 | 3 | 0 | (2+3=)5 | (5-0=)5 | (5-3=)2 |
| P0 | 7 | 0 | (5+7=)12 | (12-0=)12 | (12-7=)5 |
| P4 | 8 | 0 | (12+8=)20 | (20-0=)20 | (20-8=)12 |
| P3 | 10 | 0 | (20+10=)30 | (30-0=)30 | (30-10=)20 |
|  |  |  |  |  |  |
| Average |  |  |  | 13.80000000 | 7.80000000 |

**Gnatt Chart:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **P2** | **P1** | **P0** | **P4** | **P3** |

**0 2 5 12 20 30**

1. Consider the following example containing five processes arrive at same time.

|  |  |
| --- | --- |
| Process ID | Burst Time |
| P0 | 25 |
| P1 | 15 |
| P2 | 10 |
| P3 | 25 |
| P4 | 10 |
| P5 | 25 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Process  ID | Burst  Time | Arrival  Time | Finish Time  [Prev. Finish Time + Burst  Time] | Turn Around  Time  [FinishTime-Arrival Time] | Waiting Time  [TurnAround  Time -Burst Time] |
| P2 | 10 | 0 | (-+10=)10 | (10-0=)10 | (10-10=)0 |
| P4 | 10 | 0 | (10+10=)20 | (20-0=)20 | (20-10=)10 |
| P1 | 15 | 0 | (20+15=)35 | (35-0=)35 | (35-15=)20 |
| P0 | 25 | 0 | (35+25=)60 | (60-0=)60 | (60-25=)35 |
| P3 | 25 | 0 | (60+25=)85 | (85-0=)85 | (85-25=)60 |
| P5 | 25 | 0 | (85+25=)110 | ( 110-0=)110 | (110-25=)85 |
| Average |  |  |  | 53.33333334 | 35.00000000 |

**Gnatt Chart**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **P2** | **P4** | **P1** | **P0** | **P3** | **P5** |

**0 10 20 35 60 85 110**

### **Implementation**

**//Name:Jadhav Sahil**

**//Batch: B2**

**//PRN :2020016400783091**

**//Date :23rd July ,2021**

**// Prac-02: SJF (with no preemption)Algorithm**

**import java.util.Scanner;**

**public class P2\_SJF\_SJ**

**{**

**//defining variables**

**int burstTime[];**

**int arrivalTime[]={0};**

**String[] processId;**

**int numberOfProcess;**

**void getProcessData(Scanner input){**

**System.out.print("Enter the number of Process for Scheduling:");**

**int inputNumberOfProcess = input.nextInt();**

**numberOfProcess=inputNumberOfProcess;**

**burstTime = new int[numberOfProcess];**

**arrivalTime = new int[numberOfProcess];**

**processId = new String[numberOfProcess];**

**String st = "P";**

**for(int i=0;i<numberOfProcess;i++){**

**processId[i] = st.concat(Integer.toString(i));**

**System.out.print("Enter the burst time for Process"+(i)+":");**

**burstTime[i]=input.nextInt();}**

**}**

**void sortAccordingBurstTime(int[] at, int[] bt, String[] pid){**

**boolean swapped;**

**int temp;**

**String stemp;**

**for(int i =0;i<numberOfProcess;i++){**

**swapped=false;**

**for(int j = 0;j<numberOfProcess-i-1;j++){**

**if(bt[j]>bt[j+1]){**

**temp = bt[j];**

**bt[j] =bt[j+1];**

**bt[j+1]=temp;**

**temp = at[j];**

**at[j] =at[j+1];**

**at[j+1]=temp;**

**stemp = pid[j];**

**pid[j]=pid[j+1];**

**pid[j+1]=stemp;**

**swapped=true;**

**}**

**}**

**if(swapped==false){**

**break;**

**}**

**}**

**}**

**void shortestJobFirstNPAlgorithm(){**

**int finishTime[] = new int[numberOfProcess];**

**int bt[] = burstTime.clone();**

**int at[] = arrivalTime.clone();**

**String pid[] = processId.clone();**

**int waitingTime[] = new int[numberOfProcess];**

**int turnAroundTime[] = new int[numberOfProcess];**

**sortAccordingBurstTime(at, bt, pid);**

**finishTime[0] = at[0] + bt[0];**

**turnAroundTime[0]=finishTime[0] - at[0];**

**waitingTime[0] = turnAroundTime[0] -bt[0];**

**for(int i = 1;i<numberOfProcess;i++){**

**finishTime[i] = bt[i] + finishTime[i-1];**

**turnAroundTime[i]=finishTime[i] - at[i];**

**waitingTime[i] = turnAroundTime[i] -bt[i];**

**}**

**float sum = 0;**

**for(int n :waitingTime){**

**sum += n;**

**}**

**float averageWaitingTime = sum/ numberOfProcess;**

**sum = 0;**

**for(int n :turnAroundTime){**

**sum += n;**

**}**

**float averageTurnAroundTime = sum/ numberOfProcess;**

**System.out.println("SJF(with no preemption) Schedulling Algorithm :");**

**System.out.format("%20s%20s%20s%20s%20s%20s\n", "ProcessId", "BurstTime","ArrivalTime","FinishTime","TurnAroundTime", "WaitingTime");**

**for(int i = 0;i< numberOfProcess;i++){**

**System.out.format("%20s%20d%20d%20d%20d%20d\n", pid[i], bt[i], at[i],finishTime[i],turnAroundTime[i], waitingTime[i]);**

**}**

**System.out.format("%80s%20f%20f\n","Average", averageTurnAroundTime, averageWaitingTime);**

**}**

**public static void main(String[] args){**

**Scanner input= new Scanner(System.in);**

**P2\_SJF\_SJ obj = new P2\_SJF\_SJ();**

**obj.getProcessData(input);**

**obj.shortestJobFirstNPAlgorithm();**

**}**

**}**

### **Input:**

**Enter the number of process for scheduling:5**

**Enter the burst time for Process0:6**

**Enter the burst time for Process1:3**

**Enter the burst time for Process2:8**

**Enter the burst time for Process3:3**

**Enter the burst time for Process4:4**

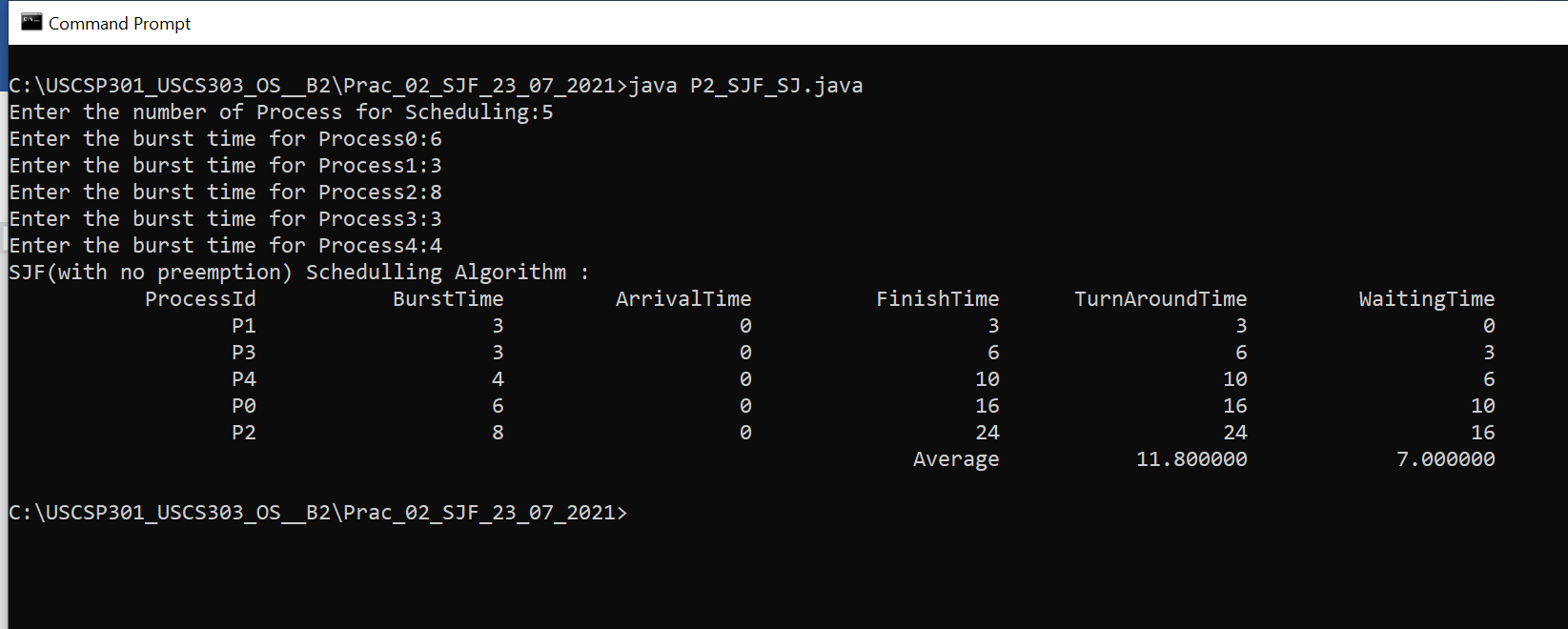
### **Output:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Process**  **ID** | **Burst**  **Time** | **Arrival**  **Time** | **Finish Time**  **[Prev. Finish Time + Burst**  **Time]** | **Turn Around**  **Time**  **[FinishTime-Arrival Time]** | **Waiting Time**  **[TurnAround**  **Time -Burst Time]** |
| **P1** | **3** | **0** | **(-+3=)3** | **(3-0=)6** | **(6-6=)0** |
| **P3** | **3** | **0** | **(3+3=)6** | **(9-0=)9** | **(9-3=)6** |
| **P4** | **4** | **0** | **(6+4=)10** | **(17-0=)17** | **(17-8=)9** |
| **P0** | **6** | **0** | **(10+6=)16** | **(20-0=)20** | **(20-3=)17** |
| **P2** | **8** | **0** | **(16+8=)24** | **(24-0=)24** | **(24-4=)20** |
|  |  |  |  |  |  |
| **Average** |  |  |  | **11.80000000** | **7.00000000** |

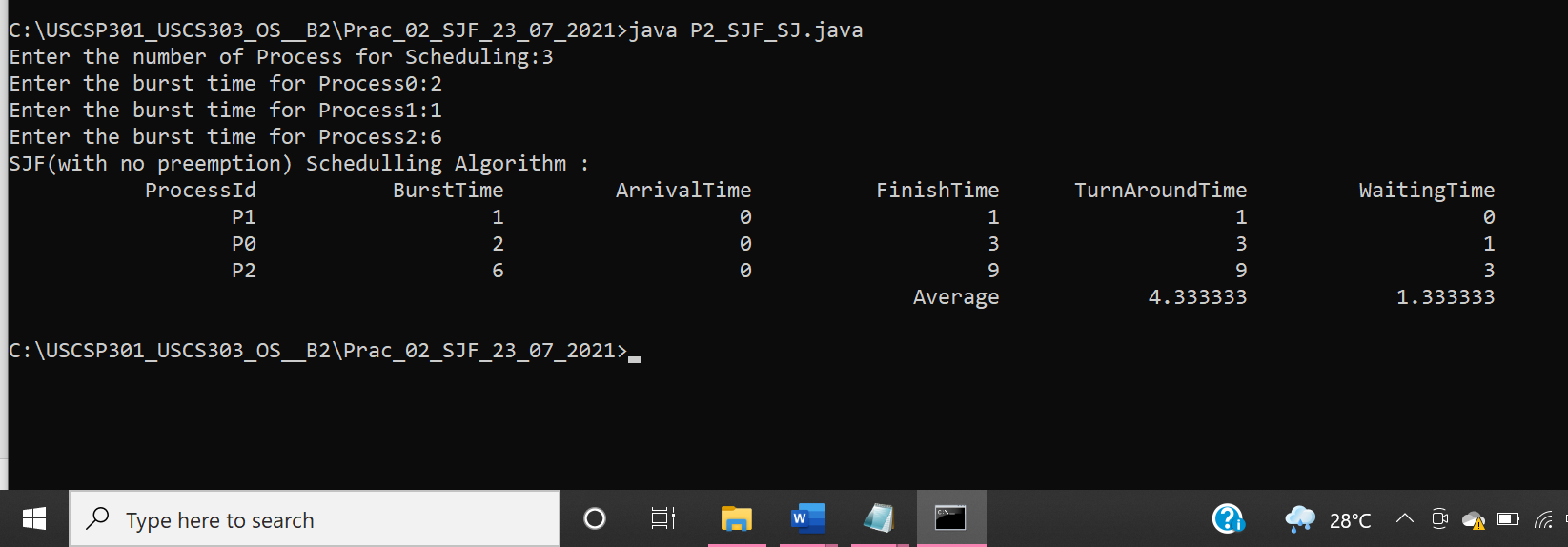
### **Sample Output(Screenshots of all the examples)** :

### 

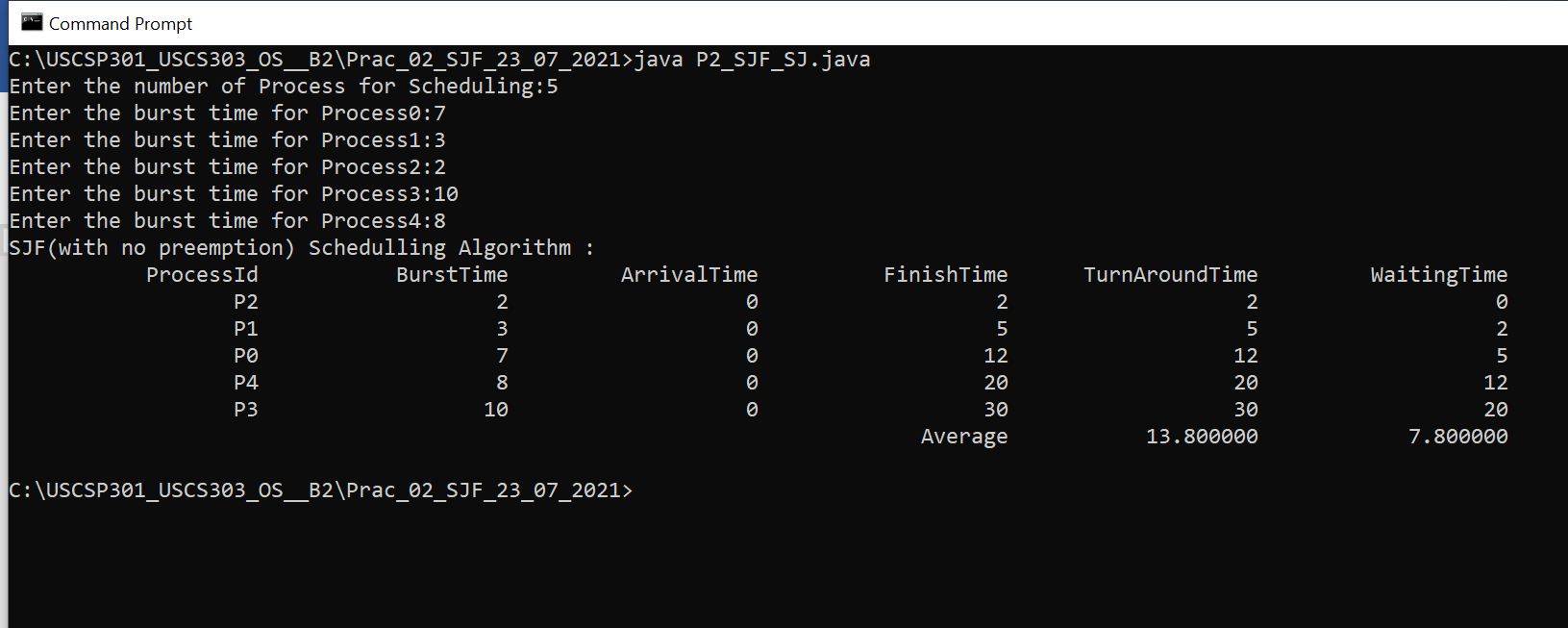
**Question1:**



**Question2:**



**Question3:**



**Question4:**

