Section → BSc 9+ 2(A) Name → Shivani Negi
Course → BSc(9+) Student I'd → 20052026
Date Page No.

Q1.) Aim → A program to implement fcfs process scheduling algorithm.

Description →

FCFS
First Come First Serve CPU Scheduling

* In fcfs scheduling
* The process which arrives first in the ready queue is firstly assigned to the CPU.
* In case of a tie, process with smaller process id is executed first.
* It is always non-preemptive in nature.

Date

Page No.

Algorithm

FCFS algorithm

STEP 1:- START

STEP 2:- Declare variable i, j, n, t, bt[MAX], at[MAX], wt[MAX], tat[MAX], temp[MAX], awt=0, atat=0.

STEP 3:- Read n, Bt[MAX], at[MAX]

STEP 4:- Temp[0] ← Bt[0]

STEP 5:- for i=0 until i<n repeat STEP 6, 7, 8, 9, 10

STEP 6:- wt[i] ← 0
tat[i] ← 0;

STEP 7:- temp[i+1] ← temp[i]+bt[i];
wt[i] ← temp[i]-at[i]

STEP 8:- tat[i] = wt[i] + bt[i]

STEP 9:- awt ← awt + wt[i]

STEP 10:- atat = atat + tat[i]

STEP 11:- PRINT: "Bt[i], at[i], wt[i], tat[i]"

STEP 12:- awt ← awt/n

STEP 13:- atat ← atat/n;

Bharti

Teacher's Signature

Date_____

TEP 14:-	PRINT:" aut "
TEP 15	PRINT:" atat "
TEP 16	STOP

Date

Page No.

C Program to implement fcfs Scheduling with arrival time

```
#include <stdio.h>
#define MAX 30

void main()
{
    int i, j, n, bt[MAX], at[MAX], wt[MAX],
    tat[MAX], temp[MAX];
    float awt=0, atat=0;

    printf("Enter the no of processes");
    scanf("%d", &n);

    printf("Enter the burst time of the proces");

    for(i=0; i<n; i++)
    scanf("%d", &bt[i]);

    printf("Enter the arrival time of the proces");

    for(i=0; i<n; i++)
    scanf("%d", &at[i]);
    temp[0] = 0;
    printf("process\t burst time\t arrival time\t
    waiting time\t turn around time");
```

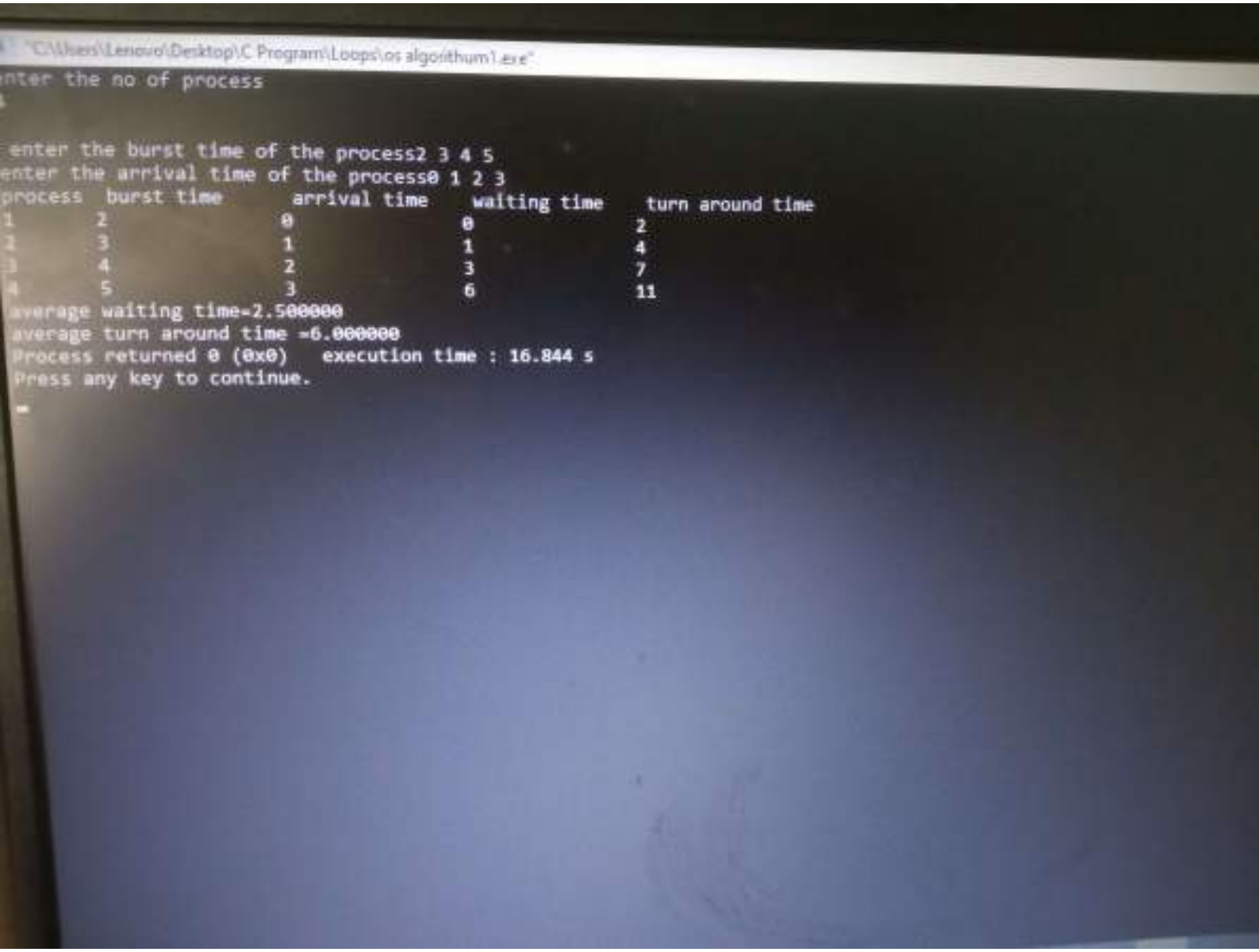
Bharti

Teacher's Signature

Date

Page No.

```
for (i=0; i<n; i++)
{
    wt[i] = 0;
    tat[i] = 0;
    temp[i+1] = temp[i] + bt[i];
    wt[i] = temp[i] - at[i];
    tat[i] = wt[i] + bt[i];
    awt = awt + wt[i];
    atat = atat + tat[i];
printf("%d\t%d\t\t%d\t\t%d\t\t%d",
        i+1, bt[i], at[i], wt[i], tat[i]);
}
    awt = awt/n;
    atat = atat/n;
printf("Average waiting time = %f\n", awt);
printf("average turn around time = %f",
                atat);
return 0;
}
```


"C:\Users\Lenovo\Desktop\C Program\Loops\os algorithum1.exe"
enter the no of process
enter the burst time of the process2 3 4 5
enter the arrival time of the process0 1 2 3
process burst time arrival time waiting time turn around time
1 2 0 0 2
2 3 1 1 4
3 4 2 3 7
4 5 3 6 11
average waiting time=2.500000
average turn around time =6.000000
Process returned 0 (0x0) execution time : 16.844 s
Press any key to continue.

Course → BSc(IT)
Section → A (A)
Date

Shivani Negi
Student I'd → 20052026
Page No

20) AIM → Write a program to implement SJF processes scheduling algorithm.

Description →

SJF
Shortest job first CPU Scheduling

* There are two types of SJF
* Preemptive SJF
* Non-Preemptive SJF

These algorithms schedule processes in the order in which the shortest job is done first. It had a minimum average waiting time.

* There are 3 factores to consider while solving SJF. They are :-

1. Burst Time
2. Average Waiting time
3. Average turnaround time.

Date

Page No.

ALGORITHM

P 1: START

P2: Declare variables i, j, n, t, p[MAX], bt[MAX], wt[MAX], tat[MAX].

STEP 3: Read n, P[MAX], bt[MAX]

EP4:- Using a for loop until i<n and repeat steps 5, 6, 7 and 8.

P5:- Using a for loop until j<n-i-1 and repeat Step 6.

P6:- Using if Condition (bt[j] > bt[j+1])

EP 7:- t = bt[j]
bt[j] = bt[j+1]
bt[j+1] = t

EP8:- t = P[j]
P[j] = p[j+1]
P[j+1] = t

Teacher's Signature

Date _____ Page No _____

P9 :- Using two nested for loops for i and j, i, j until i<n and j≤i, repeat STEPS 10, 11, 12, 13

P10 :- wt[i] = 0
tat[i] = 0

STEP 11 :- wt[i] = wt[i] + bt[i]

EP12 :- tat[i] = wt[i] + bt[i]
awt = awt + wt[i]
atat = atat + tat[i]

EPS13 :- Print P[i], bt[i], wt[i], tat[i]

EP 14 :- awt = awt/n , atat = atat/n

P15 :- Print awt, atat
P 16 :- STOP

Date

Page No.

(Shortest Job first)

```
#include <stdio.h>
#define MAX 30
int main()
{
 int i, j, n, t, p[MAX], bt[MAX], wt[MAX], tat[MAX];
 float awt=0, atat=0;

 printf("Enter the number of process\n");
  scanf("%d", &n);
 printf("Enter the process number \n");
  for(i=0; i<n; i++)
  {
     scanf("%d", &p[i]);
  }
 printf("Enter the burst time of the process \n");
  for(i=0; i<n; i++)
    scanf("%d", &bt[i]);

  for(i=0; i<n; i++)
  {
    for(j=0; j<n-i-1; j++)
    {
      if(bt[j] > bt[j+1])
      {
```

Bharti

Teacher's Signature

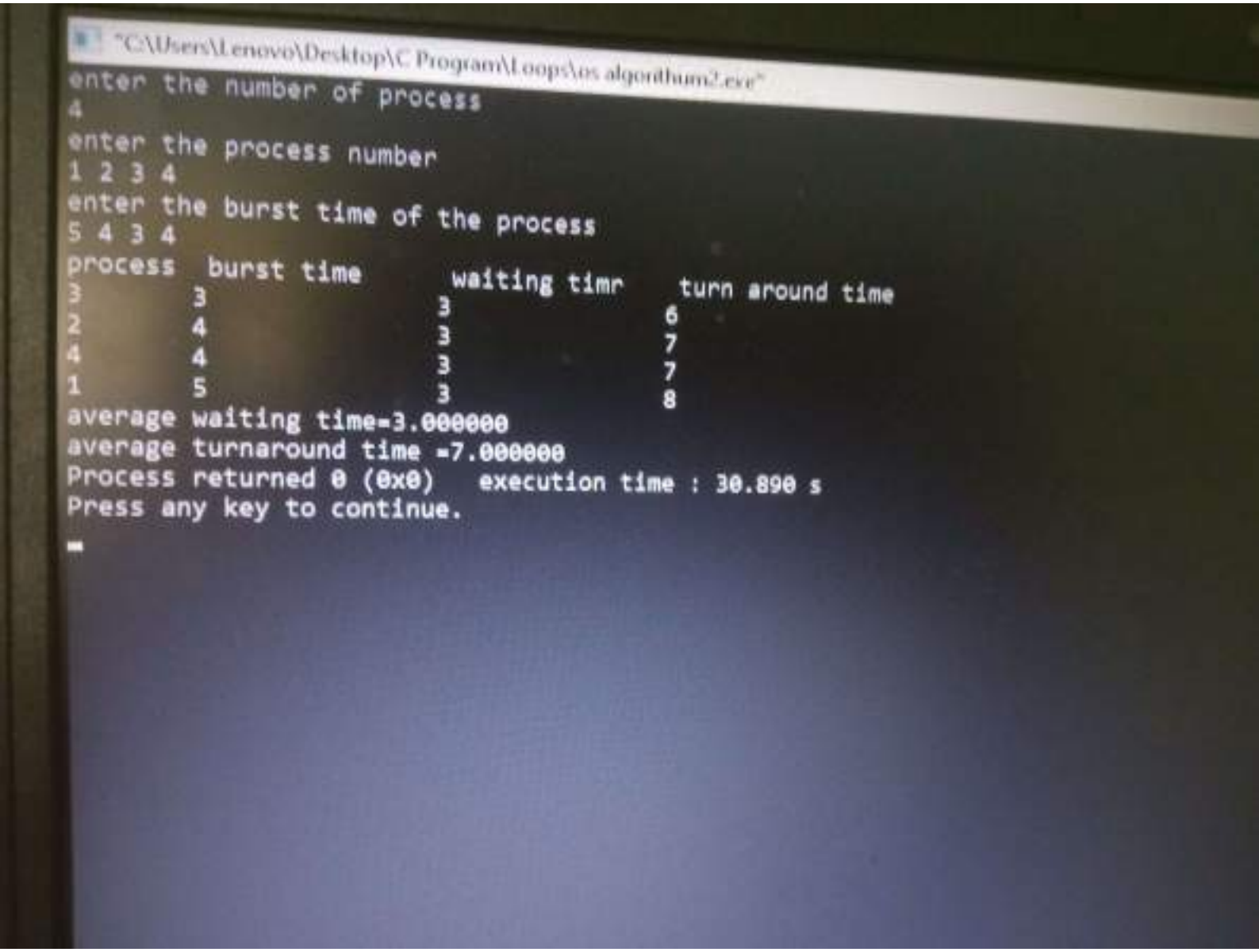
Date

Page No.

```
        t = bt[j];
        bt[j] = bt[j+1];
        bt[j+1] = t;
        t = p[j];
        p[j] = p[j+1];
        p[j+1] = t;
    }
  }
}
printf("Process\t Burst time \t Waiting time \t Turnaround
        time \n");

for(i=0; i<n; i++)
{
    wt[i] = 0;
    tat[i] = 0;
  for(j=0; j<i; j++)
  {
      wt[i] = wt[i] + bt[j]; }
      tat[i] = wt[i] + bt[i];
      awt = awt + wt[i];
      atat = atat + tat[i];
  printf("%d\t %d \t \t %d \t \t %d \n", p[i],
          bt[i], wt[i], tat[i]);

{
```

Date _____ Page No _____

```
awt = awt/n;
atat = atat/n;
printf("Average waiting time = %f\n", awt);
printf("Average Turnaround time = %f", atat);

return 0;

}
```


"C:\Users\Lenovo\Desktop\C Program\Loops\os algorithum2.exe"
enter the number of process
4
enter the process number
1 2 3 4
enter the burst time of the process
5 4 3 4
process burst time waiting timr turn around time
3 3 3 6
2 4 3 7
4 4 3 7
1 5 3 8
average waiting time=3.000000
average turnaround time =7.000000
Process returned 0 (0x0) execution time : 30.890 s
Press any key to continue.