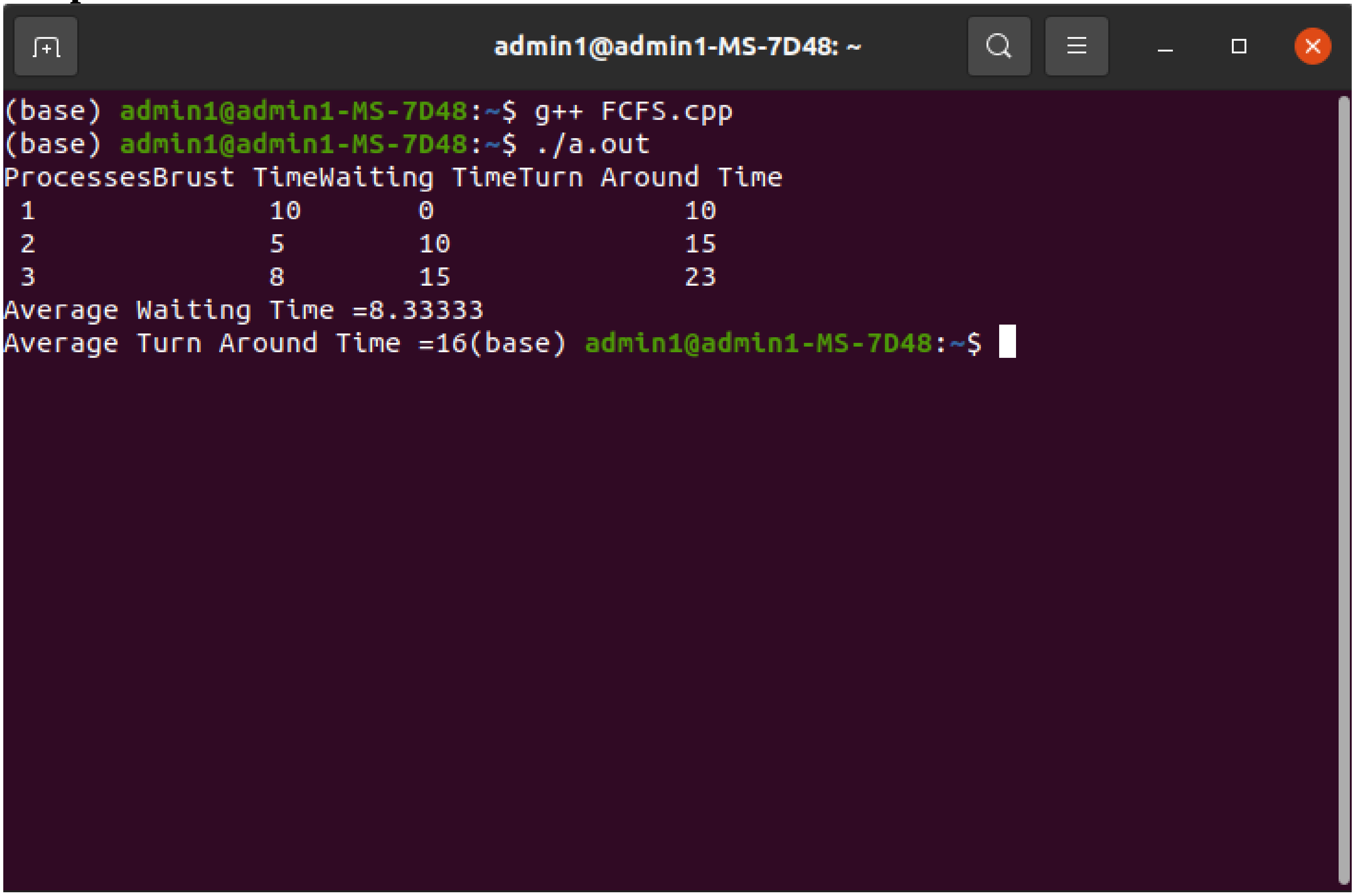
FCFS :

```
#include<iostream>
using namespace std;
void waitingtime(int processes[] , int n , int bt[] , int wt[])
{
        wt[0]=0;
        for (int i =1 ; i<n ; i++)
                wt[i] = bt[i-1]+wt[i-1];
}
void turnaroundtime (int processes[] , int n , int bt[] , int wt[] , int tat[])
{
        for(int i=0 ; i<n ; i++)
                tat[i]=wt[i]+bt[i];
}
void avgtime (int processes[] , int n , int bt[])
{
        int wt[n],tat[n],total_wt=0,total_tat=0;

        waitingtime(processes , n , bt , wt);
        turnaroundtime(processes , n , bt , wt , tat );
        cout<<"Processes"<<"Brust Time"<<"Waiting Time"<<"Turn Around Time\n";
        for (int i=0 ; i<n ; i++)
        {
                total_wt = total_wt + wt[i];
                total_tat = total_tat + tat[i];
                cout << " " << i+1 << "\t\t" << bt[i] <<"\t "
                << wt[i] <<"\t\t " << tat[i] <<endl;
        }
        cout<<"Average Waiting Time ="<<(float)total_wt/(float)n;
        cout<<"\nAverage Turn Around Time ="<<(float)total_tat/(float)n;
}
int main()
{
        int processes[] = { 1, 2 , 3};
        int n = sizeof processes/sizeof processes[0];
        int burst_time[]={10 , 5 , 8 };
        avgtime(processes , n , burst_time);
        return 0;
}
```

Output:



```
(base) admin1@admin1-MS-7D48:~$ g++ FCFS.cpp
(base) admin1@admin1-MS-7D48:~$ ./a.out
ProcessesBrust TimeWaiting TimeTurn Around Time
 1              10       0               10
 2              5        10              15
 3              8        15              23
Average Waiting Time =8.33333
Average Turn Around Time =16(base) admin1@admin1-MS-7D48:~$ 
```

Name :- ATHARV PATIL

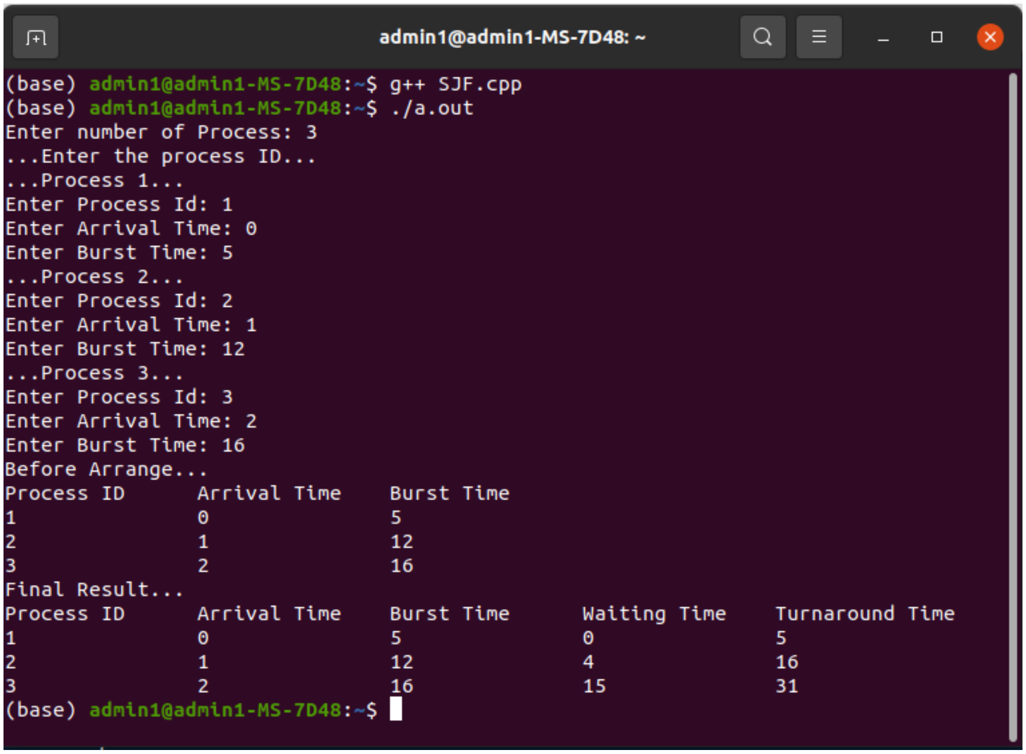
Roll no :- SEAD23133

Division :- A Batch :- A2

SJF

```
#include<iostream>
using namespace std;
int mat [10][6];
void swap(int* a, int* b)
{
    int temp = *a;
    *a = *b;
    *b = temp;
}
void arrangeArrival(int num, int mat[][6])
{
    for (int i = 0; i < num; i++)
    {
        for (int j = 0; j < num - i- 1; j++)
        {
            if (mat[j][1] > mat[j + 1][1])
            {
                for (int k = 0; k < 5; k++)
                {
                    swap(mat[j][k], mat[j + 1][k]);
                }
            }
        }
    }
}
void completionTime(int num, int mat[][6])
{
    int temp, val;
    mat[0][3] = mat[0][1] + mat[0][2];
    mat[0][5] = mat[0][3] - mat[0][1];
    mat[0][4] = mat[0][5] - mat[0][2];
    for (int i = 1; i < num; i++)
    {
        temp = mat[i - 1][3];
        int low = mat[i][2];
        for (int j = i; j < num; j++)
        {
            if (temp >= mat[j][1] && low >= mat[j][2])
            {
                low = mat[j][2];
                val=j;

            }
        }
        mat[val][3] = temp + mat[val][2];
        mat[val][5] = mat[val][3] - mat[val][1];
        mat[val][4] = mat[val][5] - mat[val][2];
        for (int k = 0; k < 6; k++)
        {
            swap(mat[val][k], mat[i][k]);
        }
    }
}
int main()
{
    int num, temp;
    cout << "Enter number of Process: ";
    cin >> num;
    cout << "...Enter the process ID...\n";
    for (int i = 0; i < num; i++)
    {
        cout << "...Process " << i + 1 << "...\n";
        cout << "Enter Process Id: ";
        cin >> mat[i][0];
        cout << "Enter Arrival Time: ";
        cin >> mat[i][1];
        cout << "Enter Burst Time: ";
        cin >> mat[i][2];
    }
    cout << "Before Arrange...\n";
    cout << "Process ID\tArrival Time\tBurst Time\n";
    for (int i = 0; i < num; i++)
    {
        cout << mat[i][0] << "\t\t" << mat[i][1] << "\t\t"<< mat[i][2] << "\n";
    }
    arrangeArrival(num, mat);
    completionTime(num, mat);
    cout << "Final Result...\n";
    cout << "Process ID\tArrival Time\tBurst Time\tWaiting ""Time\tTurnaround Time\n";
    for (int i = 0; i < num; i++)
    {
        cout << mat[i][0] << "\t\t" << mat[i][1] << "\t\t"<< mat[i][2] << "\t\t" << mat[i][4] << "\t\t"<< mat[i][5] << "\n";
    }
}
```

Output:



```
(base) admin1@admin1-MS-7D48:~$ g++ SJF.cpp
(base) admin1@admin1-MS-7D48:~$ ./a.out
Enter number of Process: 3
...Enter the process ID...
...Process 1...
Enter Process Id: 1
Enter Arrival Time: 0
Enter Burst Time: 5
...Process 2...
Enter Process Id: 2
Enter Arrival Time: 1
Enter Burst Time: 12
...Process 3...
Enter Process Id: 3
Enter Arrival Time: 2
Enter Burst Time: 16
Before Arrange...
Process ID      Arrival Time    Burst Time
1               0               5
2               1               12
3               2               16
Final Result...
Process ID      Arrival Time    Burst Time      Waiting Time    Turnaround Time
1               0               5               0               5
2               1               12              4               16
3               2               16              15              31
(base) admin1@admin1-MS-7D48:~$ 
```

Name :- ATHARV PATIL

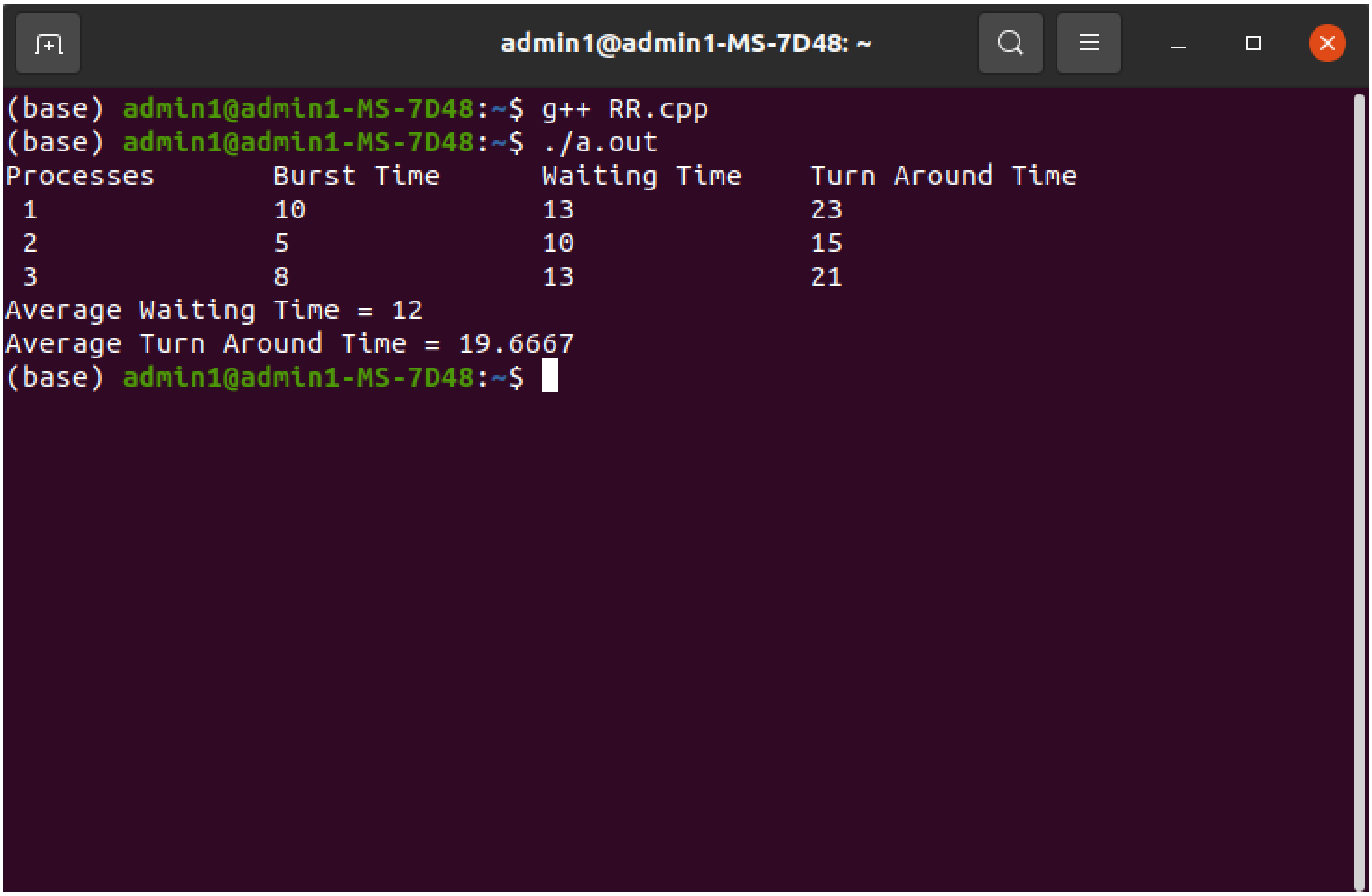
Roll no :- SEAD23133

Division :- A Batch :- A-2

RR

```
#include <iostream>
using namespace std;

void waitingtime(int processes[], int n, int wt[], int bt[], int quantum)
{
    int rem_bt[n];
    for (int i = 0; i < n; i++)
        rem_bt[i] = bt[i];

    int t = 0;  // Current time
    while (true) {
        bool done = true;
        for (int i = 0; i < n; i++)
         {
            if (rem_bt[i] > 0)
            {
                done = false;

                if (rem_bt[i] > quantum)
                {
                    t += quantum;
                    rem_bt[i] -= quantum;
                }
                else
                 {
                    t = t + rem_bt[i];
                    wt[i] = t - bt[i];
                    rem_bt[i] = 0;
                }
            }
        }
        if (done)
            break;
    }
}

void turnaroundtime(int processes[], int n, int bt[], int wt[], int tat[])
{
    for (int i = 0; i < n; i++)
        tat[i] = bt[i] + wt[i];
}

void averagetime(int processes[], int n, int bt[], int quantum)
{
    int wt[n], tat[n], totalwt = 0, totaltat = 0;

    waitingtime(processes, n, wt, bt, quantum);
    turnaroundtime(processes, n, bt, wt, tat);

    cout << "Processes\tBurst Time\tWaiting Time\tTurn Around Time\n";
    for (int i = 0; i < n; i++) {
        totalwt += wt[i];
        totaltat += tat[i];
        cout << " " << i + 1 << "\t\t" << bt[i] << "\t\t" << wt[i] << "\t\t" << tat[i] << endl;
    }

    cout << "Average Waiting Time = " << (float)totalwt / (float)n << endl;
    cout << "Average Turn Around Time = " << (float)totaltat / (float)n << endl;
}

int main()
{
    int processes[] = {1, 2, 3};
    int n = sizeof(processes) / sizeof(processes[0]);
    int bursttime[] = {10, 5, 8};
    int quantum = 2;
    averagetime(processes, n, bursttime, quantum);
    return 0;
}
```

Output:



```
(base) admin1@admin1-MS-7D48:~$ g++ RR.cpp
(base) admin1@admin1-MS-7D48:~$ ./a.out
Processes       Burst Time      Waiting Time    Turn Around Time
 1              10              13              23
 2              5               10              15
 3              8               13              21
Average Waiting Time = 12
Average Turn Around Time = 19.6667
(base) admin1@admin1-MS-7D48:~$ 
```

Name :- ATHARV PATIL

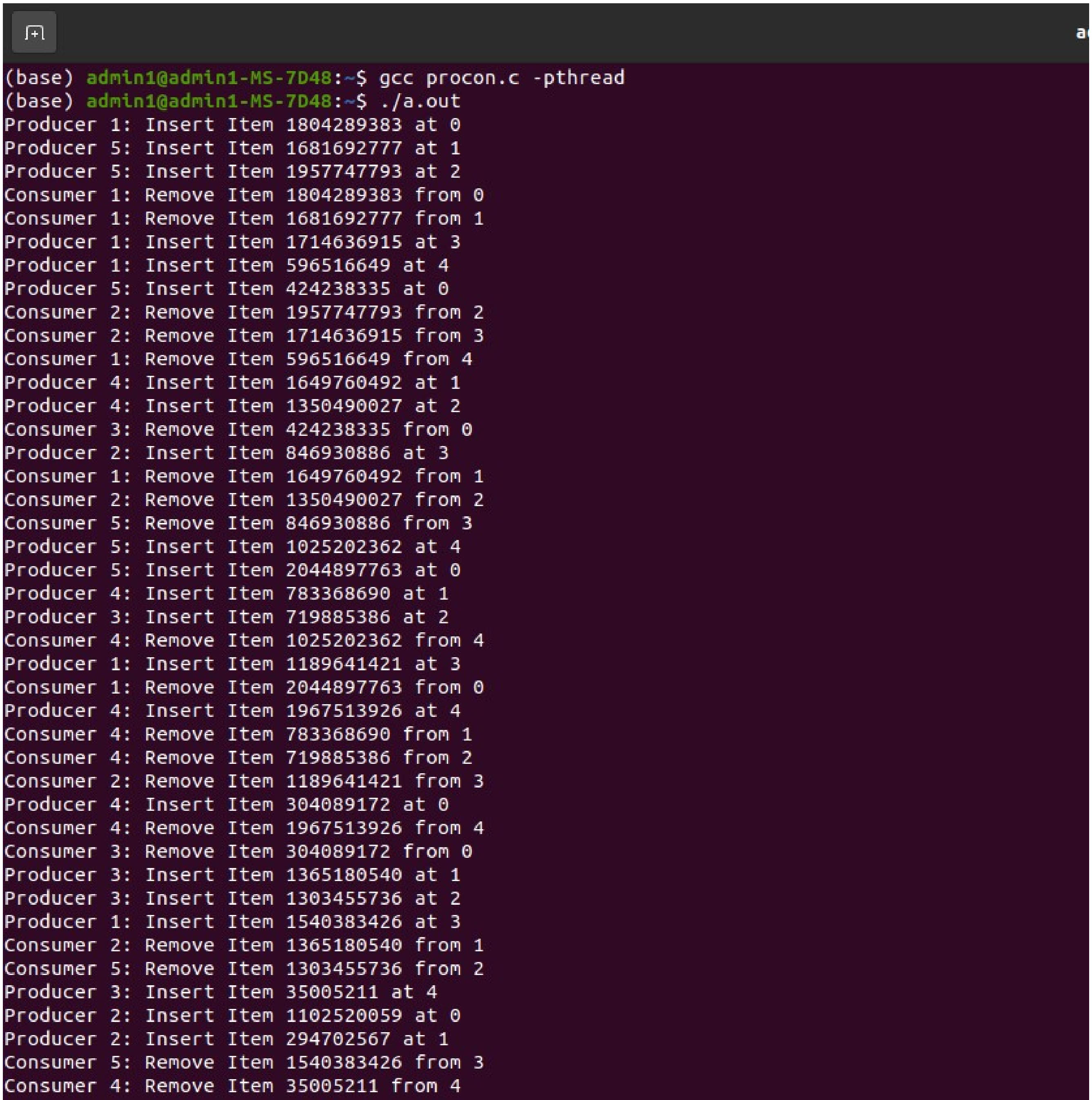
Roll no :- SEAD23133

Division :- A Batch :- A-2

PRODUCER CONSUMER PROBLEM

```
#include<pthread.h>
#include<semaphore.h>
#include<stdlib.h>
#include<stdio.h>
#define MaxItems 5
#define BufferSize 5
sem_t empty;
sem_t full;
int in = 0;
int out = 0;
int buffer[BufferSize];
pthread_mutex_t mutex;

void *producer(void *pno)
{
        int item;
        for (int i=0; i<MaxItems ; i++)
        {
                item = rand();
                sem_wait(&empty);
                pthread_mutex_lock(&mutex);
                buffer[in] = item;
                printf("Producer %d: Insert Item %d at %d\n",*((int *)pno),buffer[in],in);
                in = (in+1)%BufferSize;
                pthread_mutex_unlock(&mutex);
                sem_post(&full);
        }
}
void *consumer(void *cno)
{
        for (int i=0; i<MaxItems; i++)
        {
                sem_wait(&full);
                pthread_mutex_lock(&mutex);
                int item = buffer[out];
                printf("Consumer %d: Remove Item %d from %d\n",*((int *)cno),item,out);
                out = (out+1)%BufferSize;
                pthread_mutex_unlock(&mutex);
                sem_post(&empty);
        }
}
int main()
{
        pthread_t pro[5],con[5];
        pthread_mutex_init(&mutex,NULL);
        sem_init(&empty,0,BufferSize);
        sem_init(&full,0,0);
        int a[5]={1,2,3,4,5};
        for(int i = 0; i < 5; i++)
         {
                pthread_create(&pro[i], NULL, (void *)producer, (void *)&a[i]);
```

```
     }
     for(int i = 0; i < 5; i++)
     {
          pthread_create(&con[i], NULL, (void *)consumer, (void *)&a[i]);
     }
     for(int i = 0; i < 5; i++)
     {
          pthread_join(pro[i], NULL);
     }
     for(int i = 0; i < 5; i++)
     {
          pthread_join(con[i], NULL);
     }

    pthread_mutex_destroy(&mutex);
    sem_destroy(&empty);
    sem_destroy(&full);
    return 0;

}
```

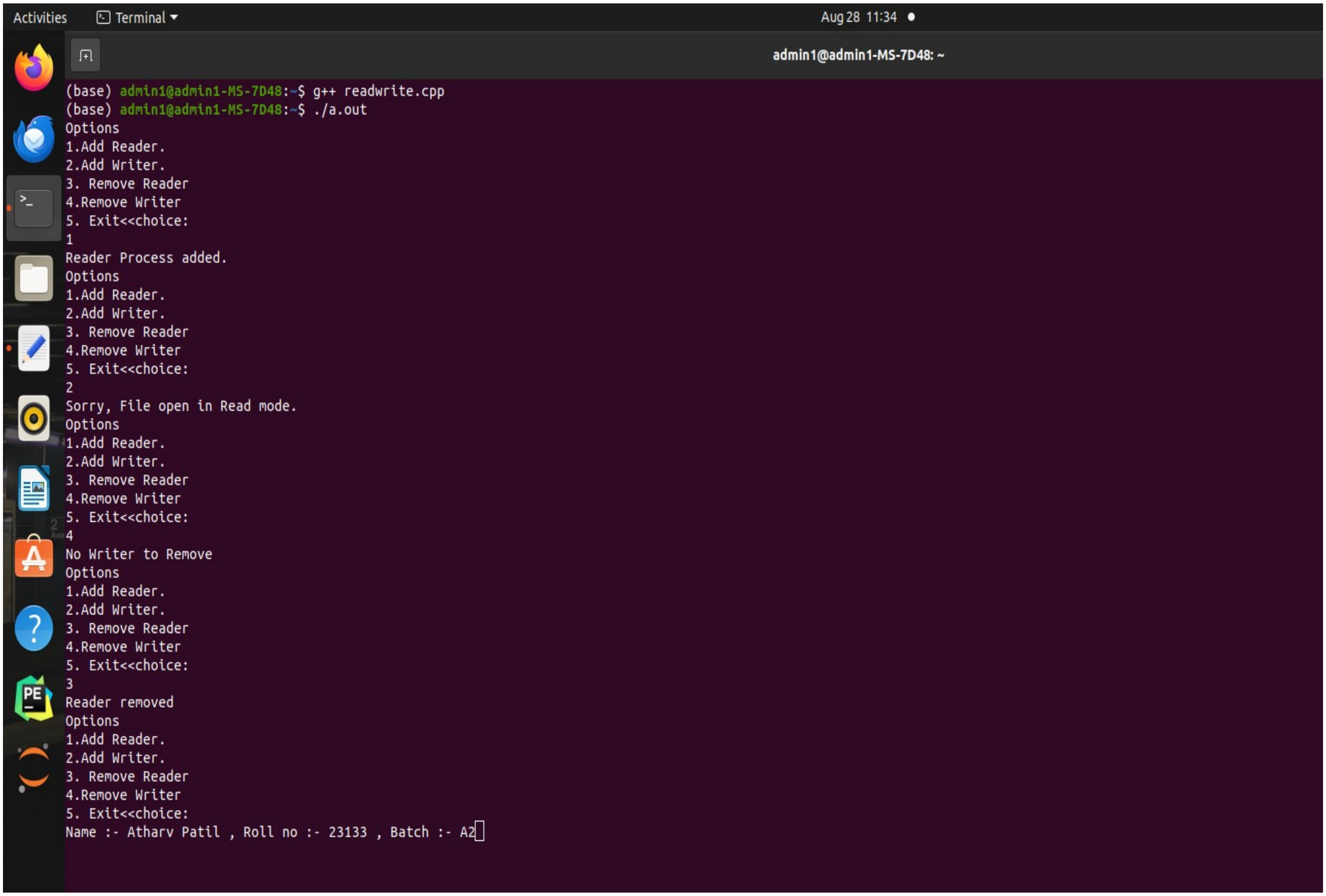


```
(base) admin1@admin1-MS-7D48:~$ gcc procon.c -pthread
(base) admin1@admin1-MS-7D48:~$ ./a.out
Producer 1: Insert Item 1804289383 at 0
Producer 5: Insert Item 1681692777 at 1
Producer 5: Insert Item 1957747793 at 2
Consumer 1: Remove Item 1804289383 from 0
Consumer 1: Remove Item 1681692777 from 1
Producer 1: Insert Item 1714636915 at 3
Producer 1: Insert Item 596516649 at 4
Producer 5: Insert Item 424238335 at 0
Consumer 2: Remove Item 1957747793 from 2
Consumer 2: Remove Item 1714636915 from 3
Consumer 1: Remove Item 596516649 from 4
Producer 4: Insert Item 1649760492 at 1
Producer 4: Insert Item 1350490027 at 2
Consumer 3: Remove Item 424238335 from 0
Producer 2: Insert Item 846930886 at 3
Consumer 1: Remove Item 1649760492 from 1
Consumer 2: Remove Item 1350490027 from 2
Consumer 5: Remove Item 846930886 from 3
Producer 5: Insert Item 1025202362 at 4
Producer 5: Insert Item 2044897763 at 0
Producer 4: Insert Item 783368690 at 1
Producer 3: Insert Item 719885386 at 2
Consumer 4: Remove Item 1025202362 from 4
Producer 1: Insert Item 1189641421 at 3
Consumer 1: Remove Item 2044897763 from 0
Producer 4: Insert Item 1967513926 at 4
Consumer 4: Remove Item 783368690 from 1
Consumer 4: Remove Item 719885386 from 2
Consumer 2: Remove Item 1189641421 from 3
Producer 4: Insert Item 304089172 at 0
Consumer 4: Remove Item 1967513926 from 4
Consumer 3: Remove Item 304089172 from 0
Producer 3: Insert Item 1365180540 at 1
Producer 3: Insert Item 1303455736 at 2
Producer 1: Insert Item 1540383426 at 3
Consumer 2: Remove Item 1365180540 from 1
Consumer 5: Remove Item 1303455736 from 2
Producer 3: Insert Item 35005211 at 4
Producer 2: Insert Item 1102520059 at 0
Producer 2: Insert Item 294702567 at 1
Consumer 5: Remove Item 1540383426 from 3
Consumer 4: Remove Item 35005211 from 4
```

READER WRITER PROBLEM

```
#include<iostream>
#include<mutex>
using namespace std;
struct semaphore
{
    int mutex;
    int rcount;
    int rwait;
    bool wrt;
};
void addreader(struct semaphore *s)
{
    if(s->mutex == 0 && s->rcount == 0)
    {
        cout<<"Sorry, File isopen in Write mode.\nNew Reader added to queue."<<endl;
        s->rwait++;
    }
    else
    {
        cout<<"Reader Process added."<<endl;
        s->rcount++;
        s->mutex--;
    }
    return;
}
void addwriter(struct semaphore *s)
{
    if(s->mutex == 1)
    {
        s->mutex--;
        s->wrt = 1;
        cout<<"\nWriter Process added."<<endl;
    }
    else if(s->wrt)
        cout<<"Sorry, Writer already operational."<<endl;
        else
            cout<<"Sorry, File open in Read mode."<<endl;
            return;
}
void removereader(struct semaphore *s)
{
    if(s->rcount == 0)
    cout<<"No readers to remove"<<endl;
    else
    {
        cout<<"Reader removed"<<endl;
        s->rcount--;
        s->mutex++;
    }
        return ;
}
```

```
void removewriter(struct semaphore *s)
{
	if(s->wrt==0)
		 cout<<"No Writer to Remove"<<endl;
	else
	{
		cout<<"Writer Removed"<<endl;
		s->mutex++;
		s->wrt=0;
		if(s->rwait!=0)
		{
			s->mutex-=s->rwait;
			s->rcount=s->rwait;
			s->rwait=0;
			cout<<"waiting Readers Added:"<<s->rcount<<endl;
		}
	}
}
int main()
{
	struct semaphore S1={1,0,0};
	while(1)
	{
		cout<<"Options"<<endl<<"1.Add Reader."<<endl<<"2.Add Writer."<<endl<<"3. Remove
Reader"<<endl<<"4.Remove Writer"<<endl<<"5. Exit<<choice:"<<endl;
	int choice;
	cin>>choice;
	switch(choice)
		{
		case 1: addreader(&S1); break;
		case 2: addwriter(&S1); break;
		case 3: removereader(&S1); break;
		case 4: removewriter(&S1); break;
		case 5: cout<<"\n\tGoodBye!";break;
		default: cout<<"\nInvalid Entry!";
		}
	}
	return 0;
}
```



```
(base) admin1@admin1-MS-7D48:~$ g++ readwrite.cpp
(base) admin1@admin1-MS-7D48:~$ ./a.out
Options
1.Add Reader.
2.Add Writer.
3. Remove Reader
4.Remove Writer
5. Exit<<choice:
1
Reader Process added.
Options
1.Add Reader.
2.Add Writer.
3. Remove Reader
4.Remove Writer
5. Exit<<choice:
2
Sorry, File open in Read mode.
Options
1.Add Reader.
2.Add Writer.
3. Remove Reader
4.Remove Writer
5. Exit<<choice:
4
No Writer to Remove
Options
1.Add Reader.
2.Add Writer.
3. Remove Reader
4.Remove Writer
5. Exit<<choice:
3
Reader removed
Options
1.Add Reader.
2.Add Writer.
3. Remove Reader
4.Remove Writer
5. Exit<<choice:
Name :- Atharv Patil , Roll no :- 23133 , Batch :- A2
```