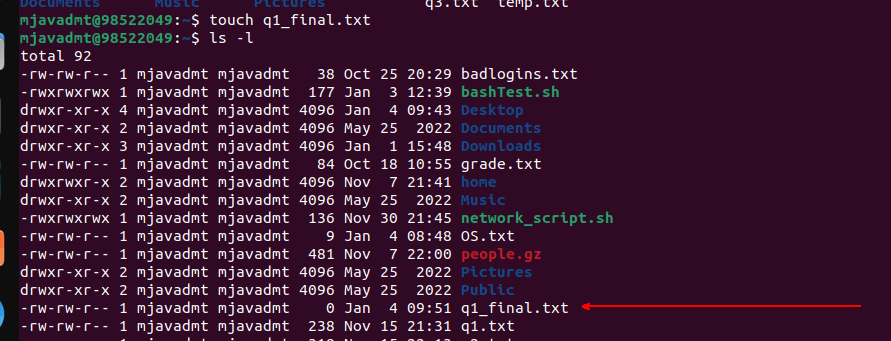
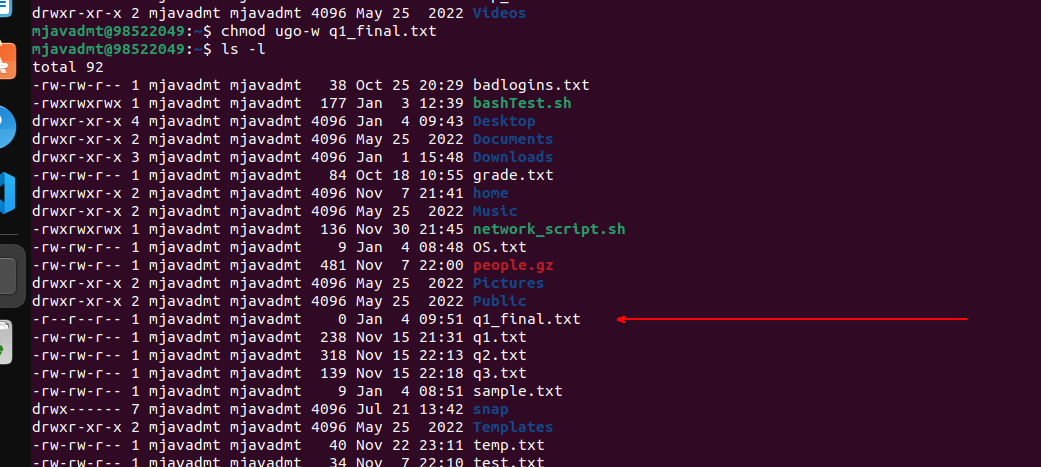
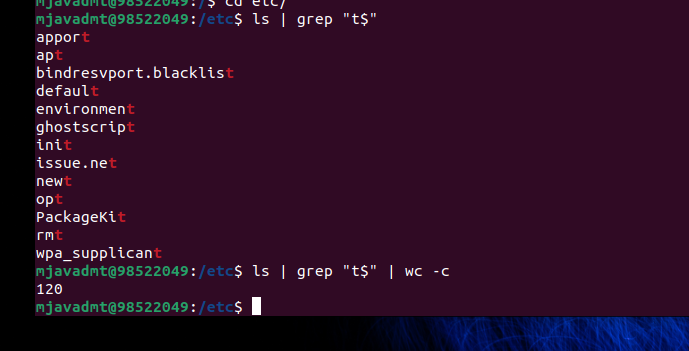
Q1.

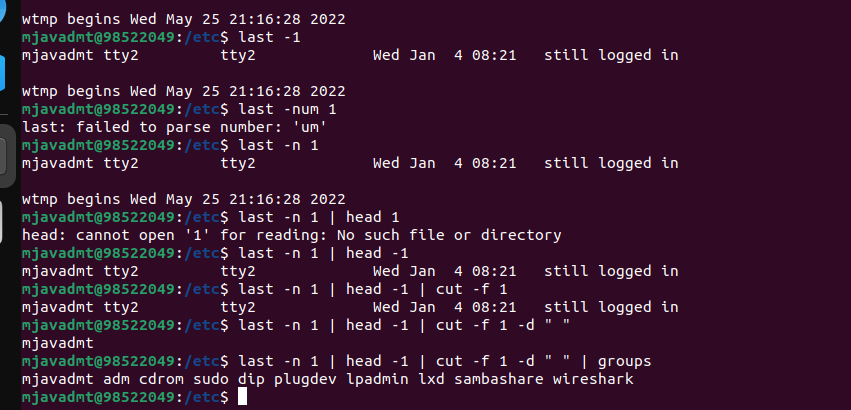




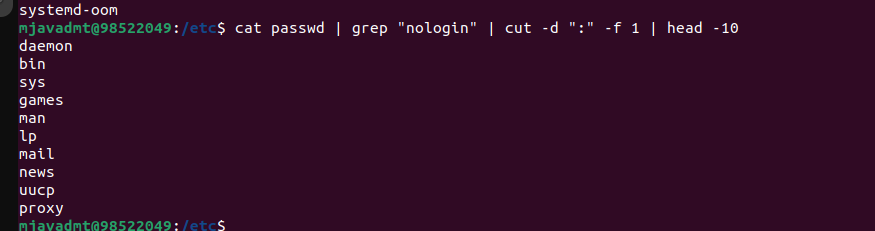
Q2.



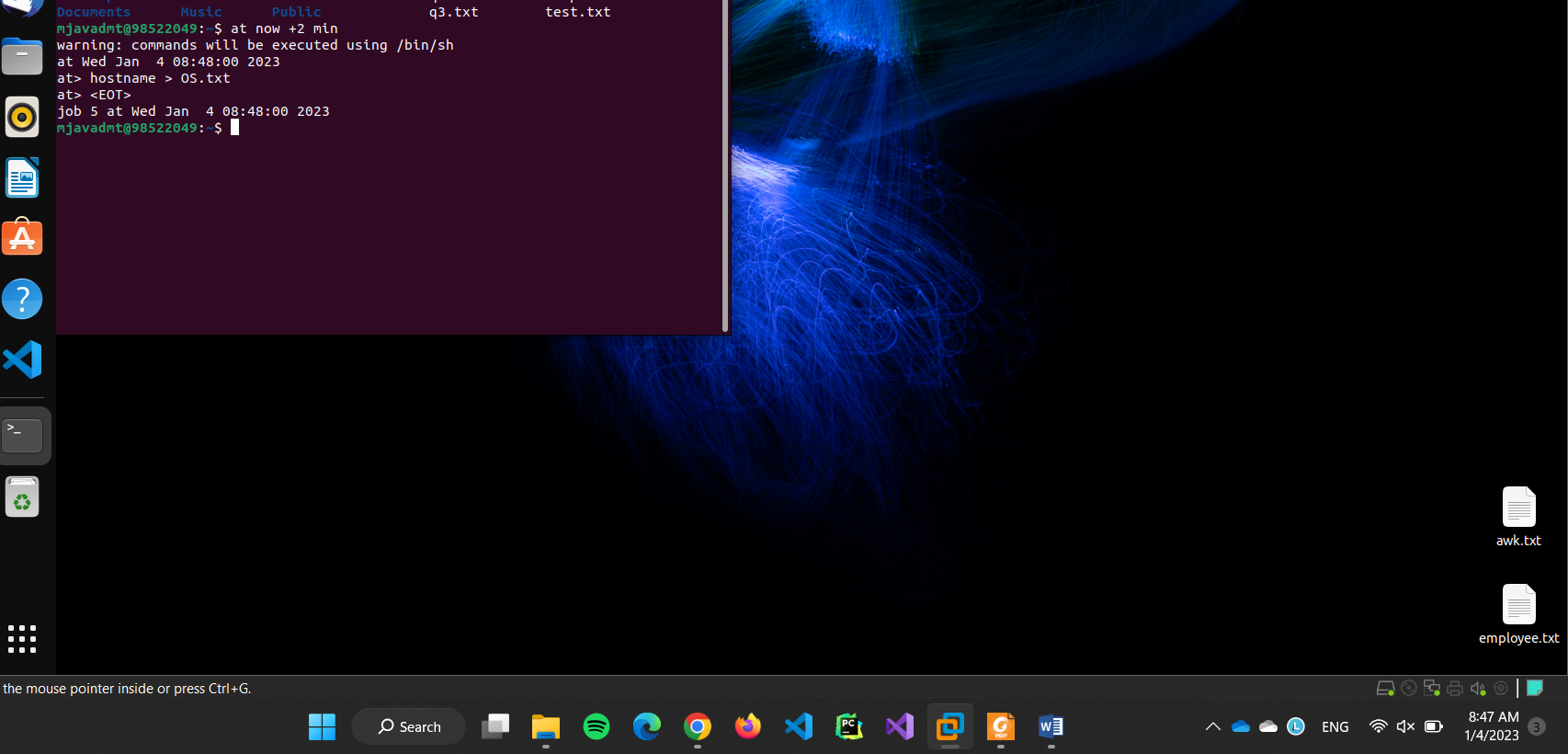
Q3.



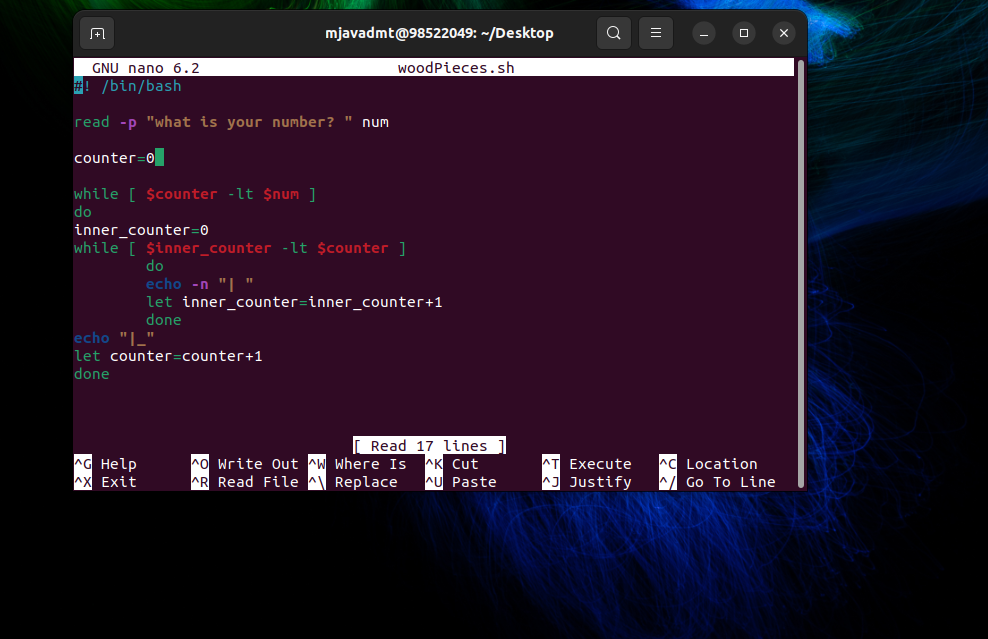
Q4.

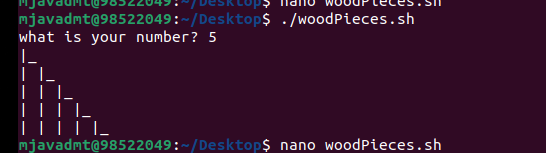


Q5.

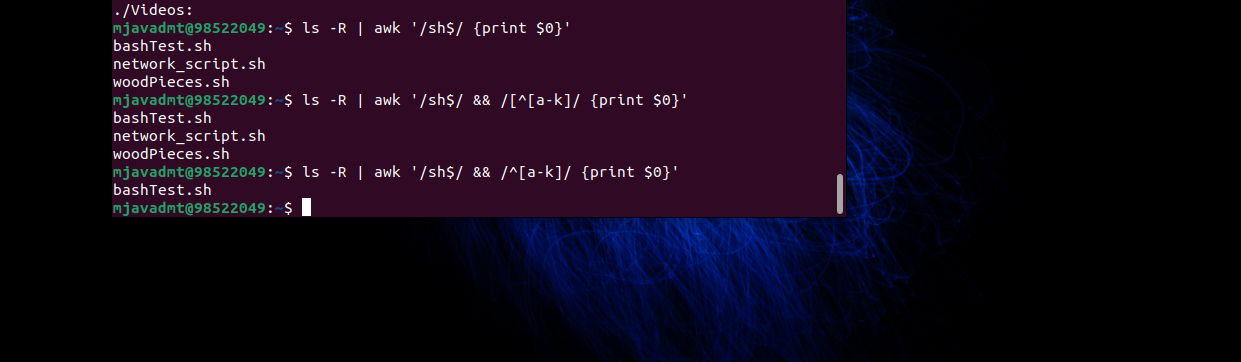


Q6.

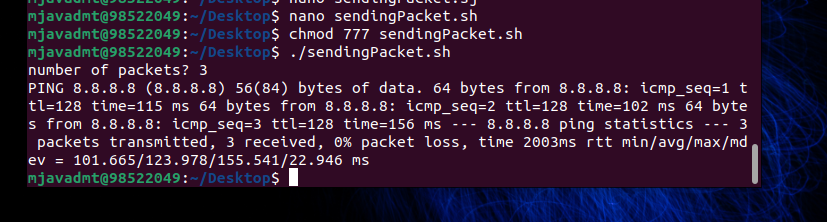


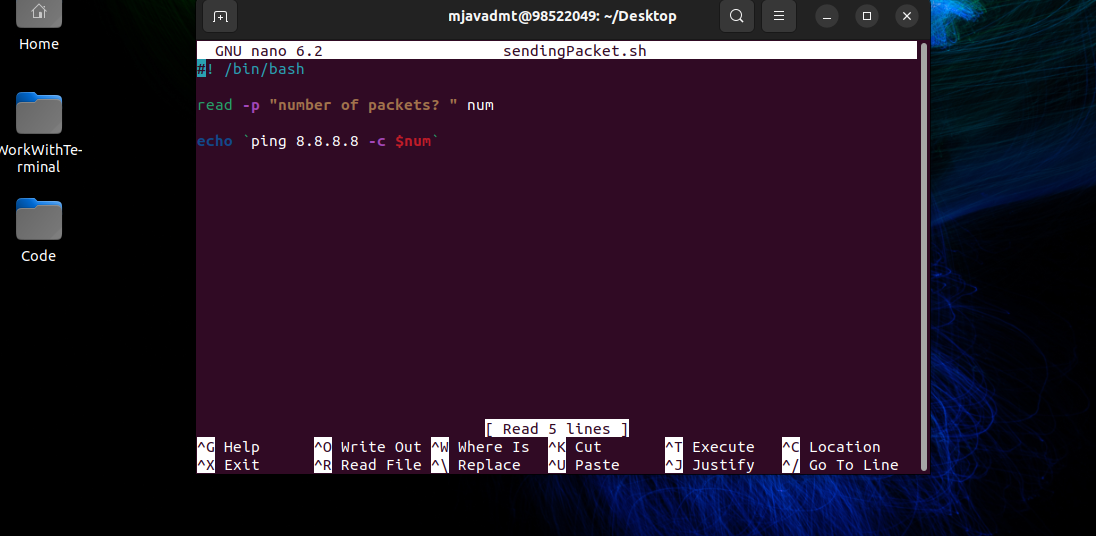


Q7.



Q8.





Q9.

#include <stdio.h>

struct process

{

    int pid, st, wt, tt, priority;

};

int main()

{

    int processes\_num;

    printf("what is number of processes? \n");

    scanf("%d", &processes\_num);

    struct process processes[processes\_num];

    for (int i = 0; i < processes\_num; i++)

    {

        // assign service time

        int service\_time;

        printf("what is service time of process with pid = %d\n", i + 1);

        scanf("%d", &service\_time);

        processes[i].st = service\_time;

        // assign priority

        int priority;

        printf("what is priority of process with pid %d\n", i+1);

        scanf("%d",&priority);

        processes[i].priority=priority;

        // assign id

        processes[i].pid = i + 1;

    }

    int i, j;

    struct process swap\_temp;

    // bubble sort

    for (i = 0; i < processes\_num - 1; i++)

    {

        for (j = 0; j < processes\_num - i - 1; j++)

        {

            if (processes[j].priority < processes[j + 1].priority)

            {

                // swapping

                swap\_temp = processes[j];

                processes[j] = processes[j + 1];

                processes[j + 1] = swap\_temp;

            }

        }

    }

    int total\_waiting = 0;

    for (int i = 0; i < processes\_num; i++)

    {

        if (i == 0)

        {

            processes[i].wt = 0;

        }

        else

        {

            processes[i].wt = processes[i - 1].tt;

        }

        total\_waiting += processes[i].wt;

        processes[i].tt = processes[i].wt + processes[i].st;

        printf("process with pid = %d, wait time = %d , service time = %d, total time= %d\n", i, processes[i].wt, processes[i].st, processes[i].tt);

    }

    double avg\_wtime = total\_waiting / processes\_num;

    printf("avg wtime = %f\n", avg\_wtime);

    return 0;

}

Q10.

