Q.) Let us consider a small but busy airport with only one run-way (shown in figure). In each time unit, one plane can land or one plane can take off, but not both. Planes arrive ready to land or to take off at random times, so at any given unit of time,the runway may be idle or a plane may be landing or taking off, and there may be several planes waiting either to land or take off. We therefore need two queues, called landing and takeoff, to hold these planes. It is better to keep a plane waiting on the ground than in the air, so a small airport allows a plane to take off only if there are no planes waiting to land. Hence, after receiving requests from new planes to land or take off, our simulation will first service the head of the queue of planes waiting to land, and only if the landing queue is empty will it allow a plane to take off. We wish to run the simulation through many units of time, and therefore, we embed the main action of the program in a loop that runs for cur-time (denoting current time) from 1 to a variable end-time.

Simulate the given scenario using and write the output for different inputs.

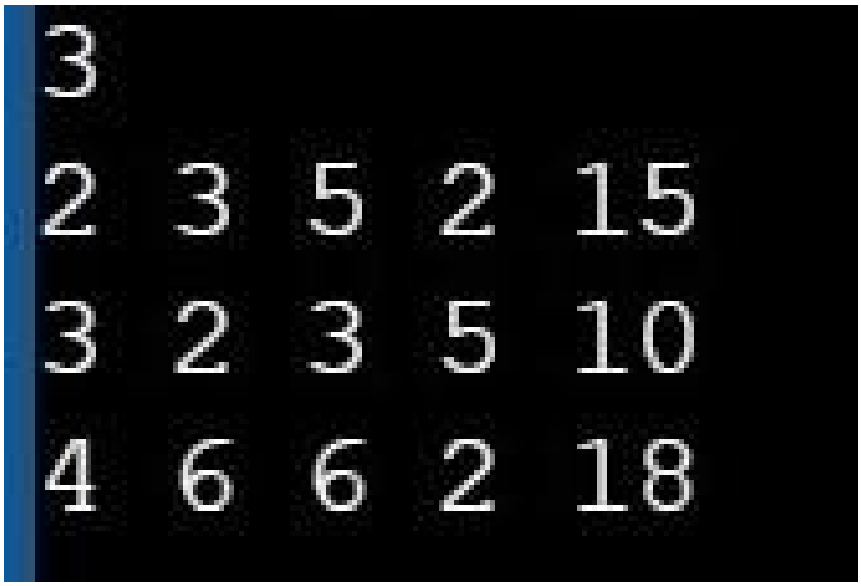
A.) (i)<u>CODE</u>:-

```
#include<stdio.h>

int main()
{
	int p,q,m,n,k;

	int cases;

	scanf("%d",&cases);

	while(cases--)
	{
	scanf("%d%d%d%d%d",&p,&q,&m,&n,&k);
	int q1=0,q2=0;
	int t=0;
	int m1;
	int i=0;
	int j=0;
	int planeland=0;
	int planetakeoff=0;
	int time[100000] = {0};

	m1=m;
```

```
while(t<=k)
{

        if(p) //when plane is moving in the air
        {
                p--;
                planeland++;
                time[i]=t;
                i++;
                t++;
                q1++;
        }

        else //when there is no plane moving in air
        {
                if(q) //plane already waiting to take off since starting
                {
                        q--;
                        planetakeoff++;
                        t++;
                }
                else // plane waiting to take off after landing
                {
                        if(t >= (time[j]+n) && q1)
                        {
                                planetakeoff++;
                                q1--;
                                t++;
                                j++;
                        }
                        else
                        t++;
                }
        }

        if(t==m1)
        {
                p++;
                m1=m1+m;
        }
}

q2=q+q1;
```

```
        printf("\n %d %d %d %d\n",planeland,planetakeoff,p,q2);

        }

        return 0;
}
```

(ii) <u>SAMPLE INPUT</u>:-



(iii) <u>SAMPLE OUTPUT</u>:-