1) Write a C program to Simulate Real time CPU scheduling Algorithms:

a) Rate - Monotonic.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
void sort(int proc[],int b[],int pt[], int n)
{
        int temp=0;
        for(int i=0;i<n; i++)
        {
              for(int j=i; j<n;j++)
              {
                     if(pt[j]<pt[i])
                     {
                            temp =pt[i];
                            pt[i] =pt[j];
                            pt[j]= temp;
                            temp = b[j];
                            b[j] = b[i];
                            b[i] = temp;
                            temp = proc[i];
                            proc[i] = proc[j];
                            proc[j] = temp;
                     }
              }
        }
}
int  gcd(int a,int b)
{
        int r;
        while(b>0)
```

```
    {
        r= a %b;
        a=b;
        b=r;
    }
    return a;
}

int lcmul(int p[], int n)
{
    int lcm=p[0];
    for( int i=1; i<n; i++)
    {
        lcm =(lcm *p[i])/gcd(lcm, p[i]);
    }
    return lcm;
}

void main()
{
    int n;
    printf("Enter the number of processes:");
    scanf("%d", &n);
    int proc[n], b[n], pt[n], rem[n];
    printf("Enter the CPU burst times:\n");
    for(int i=0; i<n; i++)
    {
        Scanf("%d", &b[i]);
        rem[i]=b[i];
    }
    printf("Enter the time periods:\n");
    for(int i=0; i<n; i++)
        Scanf("%d", &pt[i]);
```

```
for (int i=0; i<n; i++)
        proc[i] = i+1;
Sort(proc, b, pt, n);
int  l = lcmul(pt, n);
printf("LCM = %d\n", l);
printf("\n Rate Monotone  Scheduling: \n");
printf("PID\t  Burst\t period\n");
for(int  i=0; i<n; i++)
        printf("%d\t\t %d\t\t %d \n", proc[i], b[i],
               pt[i]);
double  sum = 0.0;
 for (int  i = 0;  i<n; i++)
 {
            sum += (double) b[i]/pt[i];
 }
double  rhs = n*(pow(2.0, (1.0/n)) - 1.0);
printf("\n %lf <= %lf  => %s\n", Sum, rhs,
     (Sum <= rhs)? "true": "false");
if (sum > rhs)
     exit(0);
printf(" Scheduling  occurs for %d  ms\n\n", l);
int   time = 0, prev = 0, x = 0;
while (time < l)
{
      int f = 0;
      for(int i=0; i<n; i++)
      {
            if (time % pt[i] = = 0)
                rem[i] = b[i];
            if  (rem[i]>0)
            {
                if (prev != proc[i])
                {
```

```
                printf("%dms onwards: process %d
                        running\n", time, proc[i]);
            }   prev = proc[i];

            rem[i]--;
            f=1;
          break;
            x=0;
        }
    }
    if(!f)
    {
          if(x!=1)
          {
                printf("%dms onwards: cpu is idle\n",
                        time);
                x=1;
          }
    }
    time++;
  }
}
```

Output

```
Enter the number of process: 3
Enter the CPU burst times:
3 2 2
Enter the time periods:
20 5 10
LCM = 20
Rate Monotone Scheduling:
PID      Burst      period
2          2          5
3          2          10
1          3          20
```

PAGE NO :
DATE :

0,750000 <= 0,779763 => true

Scheduling occurs for 20 ms.

0 ms onwards: process 2 running

2ms onwards: Process 3 running

4ms onwards: process 1 running

5ms onwards: process 2 running

7ms onwards: process 1 running

8ms onwards: CPU is Idle

10ms onwards: process 2 running