Student Name: **Vijay Guttula**

Student ID: **11804064 (A27)**

Email Address: **ssvijayg@gmail.com**

GitHub Link: **https://github.com/vijay-guttula/OS_CA**

Problem:

Ques2. Considering the arrival time and burst time requirement of the process the scheduler schedules the processes by interrupting the processor after every 6 units of time and does consider the completion of the process in this iteration. The scheduler than checks for the number of process waiting for the processor and allots the processor to the process but interrupting the processor every 10 unit of time and considers the completion of the processes in this iteration. The scheduler checks the number of processes waiting in the queue for the processor after the second iteration and gives the processor to the process which needs more time to complete than the other processes to go in the terminated state. The inputs for the number of requirements, arrival time and burst time should be provided by the user.

Consider the following units for reference.

Process	Arrival time	Burst time
P1	**0**	**20**
P2	**5**	**36**
P3	**13**	**19**
P4	**26**	**42**

Develop a scheduler which submits the processes to the processor in the defined scenario and compute the scheduler performance by providing the waiting time for process, turnaround time for process and average waiting time and turnaround time.

Ans: We can solve this question by using algorithm of Operating System:

1. Round Robin Scheduling Algorithm: for fixed time slice i.e for X unit and Y unit of time.

Algorithm:

```
                    time_req = 0;

            // Add time for process on left of p

            // (Scheduled before p in a round of

                    // 1 unit time slice)
```

```
                for (int i=0; i<p; i++)
                        {
                  if (arr[i] < arr[p])
                   time_req += arr[i];
                          else
                   time_req += arr[p];
                          }

               // step 2 : Add time of process p
                      time_req += arr[p];

               // Add time for process on right
                 // of p (Scheduled after p in
               // a round of 1 unit time slice)
                   for (int i=p+1; i<n; i++)
                        {
                     if (arr[i] < arr[p])
                      time_req += arr[i];
                          else
                     time_req += arr[p]-1;
                          }
```

.

Description:

To implement the above problem, we must make three iterations in which first iteration with a time slice of X units reduce the burst time of each process by X. In second iteration - ------ --- --- ------ ----- ---- waiting time of each process.

Code:

```
#include<stdio.h>
#include<conio.h>

void rr(int no,int remt[10],int Cur_t,int arT[10], int bsT[10]);

int main()
{
    int P_no, j, no, CurT, RemProc, indicator, time_quan, wait, tut, arT[10], bs
T[10], remt[10], x=1 ;

 indicator = 0;
    wait = 0;
    tut = 0;
```

```
    scanf("%d",&no);
    RemProc = no;
    printf("\nEnter the arrival time and burst time of the processes\n");
    for(P_no = 0; P_no < no; P_no++)
    {
        printf("\nProcess P%d\n", P_no + 1);
        printf("Arrival time = ");
        scanf("%d", &arT[P_no]);
        printf("Burst time = ");
        scanf("%d",&bsT[P_no]);
        remt[P_no] = bsT[P_no];
    }
    printf("The details of time quantum are as follows:\n");
    printf("The time quantum for first round is 6.\n");
    time_quan = 6;
    CurT = 0;
    for(P_no = 0; RemProc != 0 ;)
    {
        if(remt[P_no] <= time_quan && remt[P_no] >0 )
        {
            CurT += remt[P_no];
            remt[P_no] = 0;
            indicator = 1;
        }
        else if(remt[P_no] > 0)
        {
            remt[P_no] -= time_quan;
            CurT += time_quan;
        }



if(remt[P_no] == 0 && indicator == 1)

 { printf("%d",P_no);
            RemProc-- ;
            printf("P %d", P_no + 1);
            printf("\t\t\t%d", CurT - arT[P_no]);
            printf("\t\t\t%d\n", CurT - bsT[P_no] - arT[P_no]);
            wait += CurT - arT[P_no] - bsT[P_no];
            tut += CurT - arT[P_no];
            indicator = 0;

        }
        if(P_no == no - 1)
        {
            x++ ;
            if(x == 2)

  {
                P_no = 0;
                time_quan = 10;
```

```
                printf("The time quantum for second round is 10. \n");
            }
            else
            {
                break;
            }
        }
        else if(CurT >= arT[P_no + 1])
        {
            P_no++ ;
        }
        else
        {
            P_no = 0 ;
        }
    }

    rr(no, remt, CurT, arT, bsT);

    return 0;
}


void rr(int no, int remt[10], int Cur_t, int arT[10], int bsT[10])
{

    float avg_wait, avg_tut;
    int i , j, n=no, temp, btime[20], P_no[20], w_time[20], tut_t[20], total=0,

loc;


 printf("Third round with least burst time.\n");

     for(i = 0; i < n; i++)
    {btime[i] = remt[i];


w_time[i] = Cur_t - arT[i] - btime[i];
        P_no[i] = i + 1;
    }

    for(i=0;i<n;i++)
    {
        loc = i;
        for(j = i + 1; j < n; j++)
        {
            if(btime[j] < btime[loc])

 {

loc = j;
```

```
        }
        temp = btime[i];
        btime[i] = btime[loc];
        btime[loc] = temp;
        temp = P_no[i];
        P_no[i] = P_no[loc];
        P_no[loc] = temp;
    }

    for(i = 1; i < n; i++)
    {
        for(j = 0; j < i ;j++){
            w_time[i] += btime[j];
        }
        total += w_time[i];
    }


    avg_wait = (float)total / n;
    total = 0;
    printf("\nProcess\t\tBurst time\t\twaiting time\t\tTurnaround Time");
    for(i = 0; i < n; i++)
    {
        tut_t[i] = btime[i] + w_time[i];
        total = total + tut_t[i];
        printf("\nP%d\t\t\t%d\t\t\t%d\t\t\t%d", P_no[i], btime[i], w_time[i], tu
t_t[i]);
    }
    avg_tut = (float)total / n;


 printf("\n\nAverage waiting time = %f", avg_wait);

 printf("\n Average turnaround time = %f\n", avg_tut);
```

C:\Users\nnela\Documents\bnakers123.exe

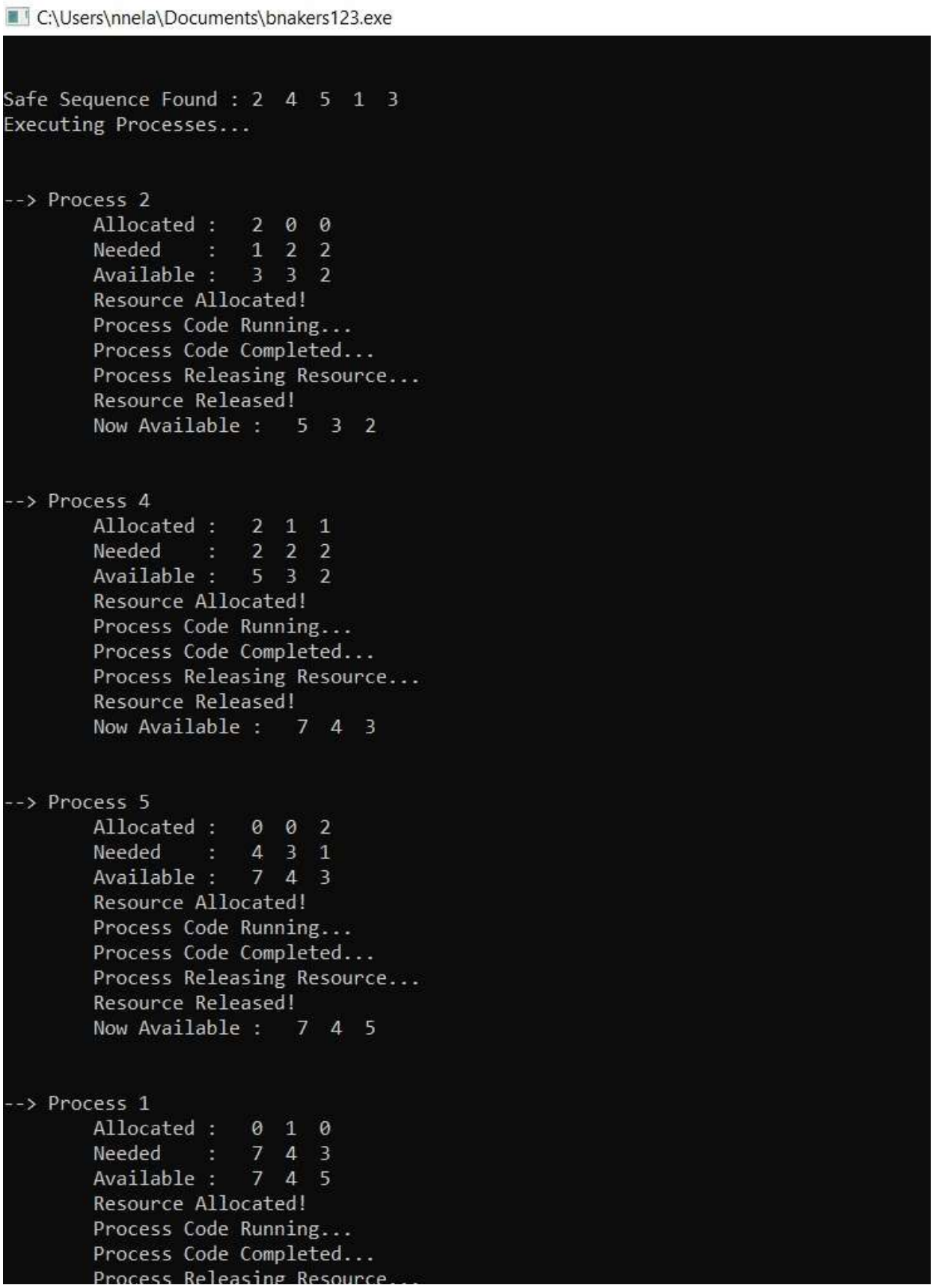
```
        Resource Released!
        Now Available :   7  4  5


--> Process 1
        Allocated :   0  1  0
        Needed    :   7  4  3
        Available :   7  4  5
        Resource Allocated!
        Process Code Running...
        Process Code Completed...
        Process Releasing Resource...
        Resource Released!
        Now Available :   7  5  5


--> Process 3
        Allocated :   3  0  2
        Needed    :   6  0  0
        Available :   7  5  5
        Resource Allocated!
        Process Code Running...
        Process Code Completed...
        Process Releasing Resource...
        Resource Released!
        Now Available :  10  5  7


All Processes Finished

--------------------------------
Process exited after 183.2 seconds with return value 1
Press any key to continue . . .
```


C:\Users\nnela\Documents\bnakers123.exe
Safe Sequence Found : 2 4 5 1 3
Executing Processes...
--> Process 2
Allocated : 2 0 0
Needed : 1 2 2
Available : 3 3 2
Resource Allocated!
Process Code Running...
Process Code Completed...
Process Releasing Resource...
Resource Released!
Now Available : 5 3 2
--> Process 4
Allocated : 2 1 1
Needed : 2 2 2
Available : 5 3 2
Resource Allocated!
Process Code Running...
Process Code Completed...
Process Releasing Resource...
Resource Released!
Now Available : 7 4 3
--> Process 5
Allocated : 0 0 2
Needed : 4 3 1
Available : 7 4 3
Resource Allocated!
Process Code Running...
Process Code Completed...
Process Releasing Resource...
Resource Released!
Now Available : 7 4 5
--> Process 1
Allocated : 0 1 0
Needed : 7 4 3
Available : 7 4 5
Resource Allocated!
Process Code Running...
Process Code Completed...

```
Enter number of processes 4

Enter the arrival time and burst time of the processes

Process P1
Arrival time = 0
Burst time = 20

Process P2
Arrival time = 5
Burst time = 36

Process P3
Arrival time = 13
Burst time = 19

Process P4
Arrival time = 26
Burst time = 42
The details of time quantum are as follows:
The time quantum for first round is 6.
The time quantum for second round is 10.
0P 1                    44                      24
Third round with least burst time.

Process         Burst time              waiting time            Turnaround Time
P1                      0                       74                      74
P3                      3                       55                      58
P2                      14                      61                      75
P4                      26                      39                      65

Average waiting time = 38.750000
 Average turnaround time = 68.000000

--------------------------------
Process exited after 78 seconds with return value 0
Press any key to continue . . .
```