INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, SURAT, KAMREJ-394919

NAME: Nakul Mantri

SEMESTER: 4th

YEAR: 2nd

ROLL NO.: UI21CS34

BRANCH: CSE

SUBJECT: OPERATING SYSTEMS (CS 401)

SUBMITTED TO: SHRADDHA PATEL & RITESH KUMAR

CODE:

```
#include <bits/stdc++.h>
using namespace std;

void print(vector<int> v)
{
   string space(1,' ');
   for(auto i:v)
      cout<<i<<space<<endl;
}
bool sortcol(vector<int> a,vector<int> b)
{
   return a[1]<b[1];
}
void fcfs(vector<vector<int>>v)
{
   int n=v.size();
   vector<int> ct(n),turn(n),wait(n);
   sort(v.begin(),v.end(),sortcol);
   int c=v[0][0];
   for(int i=0;i<n;i++)
   {
      c+=v[i][1];
      ct[i]=c;
      turn[i]=ct[i]-v[i][0];
      wait[i]=turn[i]-v[i][1];
   }cout<<endl;
   cout<<"Process No.\t"<<"AT\t\t"<<"BT\t\t"<<"CT\t\t"<<"TAT\t\t"<<"WT\t\t"<<endl;
   for(int i=0;i<n;i++)
   {
   cout<<v[i][2]<<"\t\t"<<v[i][0]<<"\t\t"<<v[i][1]<<"\t\t"<<ct[i]<<"\t\t"<<turn[i]<<"\t\t"<<wait[i]<<endl;

   }
```

```
}

void sjf(vector<vector<int>>v)
{
   int n=v.size();
   vector<int> ct(n),turn(n),wait(n);
   int c=0;int i;
   sort(v.begin(),v.end());

   priority_queue<pair<int,int>,vector<pair<int,int>>,greater<pair<int,int>>>pq;

   pq.push({v[0][1],0});
   map<int,int> m;

   while(!pq.empty())
   {
      i=pq.top().second;
      // cout<<i<<endl;
      m[i]=1;
      c+=v[i][1];
      ct[i]=c;
      turn[i]=ct[i]-v[i][0];
      wait[i]=turn[i]-v[i][1];
      pq.pop();
      if(m.size()!=n)
         for(int j=i+1;j<n;j++)
            if(v[j][0]<=c && !m[j]) pq.push({v[j][1],j});
            else break;

   }

   cout<<"Process No.\t"<<"AT\t\t"<<"BT\t\t"<<"CT\t\t"<<"TAT\t\t"<<"WT\t\t"<<endl;
    for(int i=0;i<n;i++)
```

```
    {
      cout<<v[i][2]<<"\t\t"<<v[i][0]<<"\t\t"<<v[i][1]<<"\t\t"<<ct[i]<<"\t\t"<<turn[i]<<"\t\t"<<wait[i]<<endl;

    }
}

void srtf(vector<vector<int>>v)
{
   int n=v.size();
   vector<int> ct(n),turn(n),wait(n),rem_time(n);
   int i,c=0,count=0;
   sort(v.begin(),v.end());

   for(int i=0;i<n;i++)
      rem_time[i]=v[i][1];

   priority_queue<pair<int,int>,vector<pair<int,int>>,greater<pair<int,int>>>pq;


   while(count!=n)
   {
      for(int j=0;j<n;j++)
         if(v[j][0]<=c && rem_time[j]>0)
            pq.push({rem_time[j],j});

      i=pq.top().second;
      pq.pop();
      if((rem_time[i]-1)<0)
         continue;

      c++;
      rem_time[i]--;
```

```
        if(!rem_time[i])
        {
            count++;
            ct[i]=c;
            turn[i]=ct[i]-v[i][0];
            wait[i]=turn[i]-v[i][1];
        }
    }

    cout<<"Process No.\t"<<"AT\t\t"<<"BT\t\t"<<"CT\t\t"<<"TAT\t\t"<<"WT\t\t"<<endl;
    for(int i=0;i<n;i++)
        cout<<v[i][2]<<"\t\t"<<v[i][0]<<"\t\t"<<v[i][1]<<"\t\t"<<ct[i]<<"\t\t"<<turn[i]<<"\t\t"<<wait[i]<<endl;
}




void prnp(vector<vector<int>>v)
{
    int n=v.size();
    vector<int> ct(n),turn(n),wait(n);
    int c=0;int i;
    cout<<"ENter Priorities : "<<endl;
    for(int i=0;i<n;i++)
        cin>>v[i][3];
    sort(v.begin(),v.end());

    priority_queue<pair<int, int>,vector< pair<int, int>>,function<bool( pair<int, int>,  pair<int, int>)>>pq =
        priority_queue< pair<int, int>,  vector< pair<int, int>>,
            function<bool( pair<int, int>,  pair<int, int>)>>(
            []( pair<int, int> a,  pair<int, int> b) {
                if (a.first != b.first) {
                    return a.first < b.first;
                } else {
```

```
            return a.second > b.second;
        }
    });
// priority_queue<pii>pq;
vector<pair<int,int>> vv;
//Stores BT and index of vec have<int>g that BT after Sorting

pq.push({v[0][3],0});
map<int,int> m,vis;

while(!pq.empty())
{
   i=pq.top().second;

   // cout<<i+1<<endl;
   m[i]=1;
   c+=v[i][1];

   ct[i]=c;
   turn[i]=ct[i]-v[i][0];
   wait[i]=turn[i]-v[i][1];

   pq.pop();

   if(vis.size()!=n)
   {
      for(int j=i+1;j<n;j++)
         if(v[j][0]<=c && !vis[j]) {pq.push({v[j][3],j});vis[j]=1;}
         else break;
   }
}

cout<<"Process No.\t"<<"AT\t"<<"BT\t"<<"Priority\t"<<"CT\t"<<"TAT\t"<<"WT\t"<<endl;
```

```
    for(int i=0;i<n;i++)
    {
cout<<v[i][2]<<"\t\t"<<v[i][0]<<"\t"<<v[i][1]<<"\t"<<v[i][3]<<"\t\t"<<ct[i]<<"\t"<<turn[i]<<"\t"<<wait[i]<<endl;

    }
}

void prp(vector<vector<int>>v)
{
    int n=v.size();
    vector<int> ct(n),turn(n),wait(n),rem_time(n);
    int c=0;int i;
    cout<<"Enter Priorities : "<<endl;
    for(int i=0;i<n;i++)
        cin>>v[i][3];

    sort(v.begin(),v.end());

    for(int i=0;i<n;i++)
        rem_time[i]=v[i][1];

    priority_queue<pair<int, int>,vector< pair<int, int>>,function<bool( pair<int, int>,  pair<int, int>)>> pq =
        priority_queue< pair<int, int>,  vector< pair<int, int>>,
            function<bool( pair<int, int>,  pair<int, int>)>>(
            []( pair<int, int> a,  pair<int, int> b) {
                if (a.first != b.first) {
                    return a.first < b.first;
                } else {
                    return a.second > b.second;
                }
            });
```

```
    // pq.push({v[0][3],0});
    map<int,int> m,vis;
    int count=0;
    while(count!=n)
    {
        for(int j=0;j<n;j++)
            if(v[j][0]<=c && rem_time[j]>0)
                pq.push({v[j][3],j});

        i=pq.top().second;
        pq.pop();
        if((rem_time[i]-1)<0)
            continue;
        c++;
        rem_time[i]--;

        if(!rem_time[i])
        {
            count++;
            ct[i]=c;
            turn[i]=ct[i]-v[i][0];
            wait[i]=turn[i]-v[i][1];
        }
    }

    cout<<"Process No.\t"<<"AT\t"<<"BT\t"<<"Priority\t"<<"CT\t"<<"TAT\t"<<"WT\t"<<endl;
    for(int i=0;i<n;i++)
    {
cout<<v[i][2]<<"\t\t"<<v[i][0]<<"\t"<<v[i][1]<<"\t"<<v[i][3]<<"\t\t"<<ct[i]<<"\t"<<turn[i]<<"\t"<<wait[i]<<endl;

    }
}
```

```
void rr(vector<vector<int>>v)
{
   int n=v.size();
   vector<int> ct(n),turn(n),wait(n),rem_time(n);
   int i,quan,c=0;
   cout<<"Enter the quantum for round robin="<<endl;
   cin>>quan;
   sort(v.begin(),v.end());

   for(int i=0;i<n;i++)
      rem_time[i]=v[i][1];

   queue<pair<int,int>>pq;



   map<int,int> vis;
   int count=0;
   pq.push({rem_time[0],0});
   vis[0]=1;

   while(count!=n)
   {

      i=pq.front().second;
      pq.pop();

      if(rem_time[i]>quan)
      {
         c+=quan;rem_time[i]-=quan;
      }
      else
      {
```

```
            c+=rem_time[i];
            rem_time[i]=0;
        }

        for(int j=0;j<n;j++)
            if(v[j][0]<=c && rem_time[j]>0 && !vis[j])
                {pq.push({rem_time[j],j});vis[j]=1;}

        if(rem_time[i])
            pq.push({rem_time[i],i});

        queue<pair<int,int>>qq=pq;

        if(!rem_time[i])
        {
            count++;
            ct[i]=c;
            turn[i]=ct[i]-v[i][0];
            wait[i]=turn[i]-v[i][1];
        }



    }

    cout<<"Process No.\t"<<"AT\t"<<"BT\t"<<"CT\t"<<"TAT\t"<<"WT\t"<<endl;
    for(int i=0;i<n;i++)
    {
    cout<<v[i][2]<<"\t\t"<<v[i][0]<<"\t"<<v[i][1]<<"\t"<<ct[i]<<"\t"<<turn[i]<<"\t"<<wait[i]<<endl;

    }
}
```

```
int main()
{
   cout<<" Enter Number of Processes : ";
   int n;
   cin>>n;
   vector<vector<int>>ab(n,vector<int>(4));
   cout<<"Enter ArrivalTime & Burst Time: ";
   for(int i=0;i<n;i++)
      {cin>>ab[i][0]>>ab[i][1];ab[i][2]=i+1;}

   while(1)
   {
   cout<<"Enter 0(Exit),1(FCFS),2(SJF),3(SRTF),4(Preemptive Priority),5(Non-Preemptive Priority),6(Round
Robin) : ";
   int whichalgo;
   cin>>whichalgo;
   switch(whichalgo)
   {
      case 1:
      {
      fcfs(ab);break;
      }
      case 2:
      {
      sjf(ab);break;
      }
      case 3:
      {
      srtf(ab);break;
```

```
        }
        case 4:
        {
        prp(ab);break;
        }
        case 5:
        {
        prnp(ab);break;
        }
        case 6:
        {
        rr(ab);break;
        }
        case 0:
        {
           exit(0);break;
        }
        default:
        {
           cout<<"Enter Valid Number for Algorithm";
        }
    }
    }
}
```

OUTPUT:

```
 Enter Number of Processes : 4
Enter ArrivalTime & Burst Time: 0 2
1 5
2 10
3 15
Enter 0(Exit),1(FCFS),2(SJF),3(SRTF),4(Preemptive Priority),5(Non-Preemptive Priority),6(Round Robin) : 1

Process No.     AT              BT              CT              TAT             WT
1               0               2               2               2               0
2               1               5               7               6               1
3               2               10              17              15              5
4               3               15              32              29              14
Enter 0(Exit),1(FCFS),2(SJF),3(SRTF),4(Preemptive Priority),5(Non-Preemptive Priority),6(Round Robin) : 2
Process No.     AT              BT              CT              TAT             WT
1               0               2               2               2               0
2               1               5               7               6               1
3               2               10              27              25              15
4               3               15              42              39              24
Enter 0(Exit),1(FCFS),2(SJF),3(SRTF),4(Preemptive Priority),5(Non-Preemptive Priority),6(Round Robin) : 3
Process No.     AT              BT              CT              TAT             WT
1               0               2               2               2               0
2               1               5               7               6               1
3               2               10              17              15              5
4               3               15              32              29              14
Enter 0(Exit),1(FCFS),2(SJF),3(SRTF),4(Preemptive Priority),5(Non-Preemptive Priority),6(Round Robin) : 4
Enter Priorities :
1 2 3 4
Process No.     AT      BT      Priority        CT      TAT     WT
1               0       2       1               32      32      30
2               1       5       2               31      30      25
```

```
Enter Priorities :
1 2 3 4
Process No.     AT      BT      Priority        CT      TAT     WT
1               0       2       1               32      32      30
2               1       5       2               31      30      25
3               2       10      3               27      25      15
4               3       15      4               18      15      0
Enter 0(Exit),1(FCFS),2(SJF),3(SRTF),4(Preemptive Priority),5(Non-Preemptive Priority),6(Round Robin) : 5
ENter Priorities :
4 3 2 1
Process No.     AT      BT      Priority        CT      TAT     WT
1               0       2       4               2       2       0
2               1       5       3               7       6       1
3               2       10      2               17      15      5
4               3       15      1               32      29      14
Enter 0(Exit),1(FCFS),2(SJF),3(SRTF),4(Preemptive Priority),5(Non-Preemptive Priority),6(Round Robin) : 6
Enter the quantum for round robin=
2
Process No.     AT      BT      CT      TAT     WT
1               0       2       2       2       0
2               1       5       15      14      9
3               2       10      25      23      13
4               3       15      32      29      14
Enter 0(Exit),1(FCFS),2(SJF),3(SRTF),4(Preemptive Priority),5(Non-Preemptive Priority),6(Round Robin) :
```