PROGRAM-1

Write c program to simulate cpu scheduling algorithm
1. FCFS
2. SJF (preemptive and non-preemptive)

Code for FCFS:

```
//fcfs cpu scheduling
#include<stdio.h>

struct Process{
   int id, at, bt, ct, tat, wt;
};

void fcfs(struct Process p[], int n){
   int ttt=0, twt=0;
   for(int i=0;i<n-1;i++){
      for(int j=i+1;j<n;j++){
         if(p[i].at>p[j].at){
            struct Process temp=p[i];
            p[i]=p[j];
            p[j]=temp;
         }
      }
   }
   p[0].ct=p[0].at+p[0].bt;
   p[0].tat=p[0].ct-p[0].at;
   ttt+=p[0].tat;
   p[0].wt=0;
   for(int i=1;i<n;i++){
      if(p[i-1].ct>p[i].at)
         p[i].ct=p[i-1].ct+p[i].bt;
      else
         p[i].ct=p[i].at+p[i].bt;
      p[i].tat=p[i].ct-p[i].at;
      p[i].wt=p[i].tat-p[i].bt;
      ttt+=p[i].tat;
      twt+=p[i].wt;
   }
```

```
    printf("FCFS Scheduling:\n");
    printf("Process\tAT\tBT\tCT\tTAT\tWT\n");
    for(int i=0;i<n;i++){
        printf("P%d\t%d\t%d\t%d\t%d\t%d\n", p[i].id, p[i].at, p[i].bt, p[i].ct, p[i].tat, p[i].wt);
    }
    printf("\nAverage TAT: %.2f units", (float)ttt/n);
    printf("\nAverage WT: %.2f units", (float)twt/n);
}

int main(){
    int n;
    printf("Enter the number of processes:");
    scanf("%d", &n);
    struct Process p[n];
    printf("\nEnter the arrival time and burst time for process:\n");
    for(int i=0;i<n;i++){
        p[i].id=i+1;
        printf("Process %d: ", i+1);
        scanf("%d %d", &p[i].at, &p[i].bt);
    }
    fcfs(p,n);
    return 0;
}
```

Output:

```
Enter the number of processes:4

Enter the arrival time and burst time for process:
Process 1: 0 7
Process 2: 8 3
Process 3: 3 4
Process 4: 5 6
FCFS Scheduling:
Process AT      BT      CT      TAT     WT
P1      0       7       7       7       0
P3      3       4       11      8       4
P4      5       6       17      12      6
P2      8       3       20      12      9

Average TAT: 9.75 units
Average WT: 4.75 units
Process returned 0 (0x0)   execution time : 14.162 s
Press any key to continue.
```

Code for SJF (Preemptive and Non-Preemptive)

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

struct Process{
   int id, at, bt, ct, tat, wt, remt;
};

void non_preemptive(struct Process p[], int n){
   int comp=0, curr=0, ttt=0, twt=0, minidx=-1, mbt=INT_MAX;
   int iscompleted[n];

   for(int i=0;i<n;i++)
      iscompleted[i]=0;

   while(comp<n){
      mbt=INT_MAX;
      minidx=-1;

      for(int i=0;i<n;i++){
         if(!iscompleted[i] && p[i].at<=curr){
            if(p[i].bt<mbt){
               mbt=p[i].bt;
               minidx=i;
            }
         }
      }

      if(minidx==-1){
         curr++;
         continue;
      }

      iscompleted[minidx]=1;
      comp++;
      p[minidx].ct=p[minidx].bt+curr;
      p[minidx].tat=p[minidx].ct-p[minidx].at;
      p[minidx].wt=p[minidx].tat-p[minidx].bt;
```

```
        ttt+=p[minidx].tat;
        twt+=p[minidx].wt;
        curr=p[minidx].ct;
    }

    printf("SJF Non Preemptive Scheduling:\n");
    printf("Process\tAT\tBT\tCT\tTAT\tWT\n");
    for(int i=0;i<n;i++){
        printf("P%d\t%d\t%d\t%d\t%d\t%d\n", p[i].id, p[i].at, p[i].bt, p[i].ct, p[i].tat, p[i].wt);
    }
    printf("\nAverage TAT: %.2f units", (float)ttt/n);
    printf("\nAverage WT: %.2f units", (float)twt/n);
}

void preemptive(struct Process p[], int n){
    int comp=0, curr=0, ttt=0, twt=0, minidx, mrt;
    for(int i=0;i<n;i++){
        p[i].remt=p[i].bt;
    }
    while(comp<n){
        minidx=-1;
        mrt=INT_MAX;

        for(int i=0;i<n;i++){
            if(p[i].at<=curr && p[i].remt>0){
                if(p[i].remt<mrt){
                    mrt=p[i].remt;
                    minidx=i;
                }
                else if(p[i].remt==mrt && p[i].at<p[minidx].at){
                    minidx=i;
                }
            }
        }

        if(minidx==-1){
            curr++;
            continue;
        }
```

```
            p[minidx].remt--;
            curr++;

            if(p[minidx].remt==0){
                comp++;
                p[minidx].ct=curr;
                p[minidx].tat=p[minidx].ct-p[minidx].at;
                p[minidx].wt=p[minidx].tat-p[minidx].bt;
                ttt+=p[minidx].tat;
                twt+=p[minidx].wt;
            }
        }
        printf("SJF Preemptive Scheduling:\n");
        printf("Process\tAT\tBT\tCT\tTAT\tWT\n");
        for(int i=0;i<n;i++){
            printf("P%d\t%d\t%d\t%d\t%d\t%d\n", p[i].id, p[i].at, p[i].bt, p[i].ct, p[i].tat, p[i].wt);
        }
        printf("\nAverage TAT: %.2f units", (float)ttt/n);
        printf("\nAverage WT: %.2f units", (float)twt/n);
    }

    int main(){
        int n, c;
        printf("Enter the number of processes:");
        scanf("%d", &n);
        struct Process p[n];
        printf("Enter choice:\n1.Non-preemptive\n2.Preemptive\n");
        scanf("%d", &c);
        printf("\nEnter the arrival time and burst time for process:\n");
        for(int i=0;i<n;i++){
            p[i].id=i+1;
            printf("Process %d: ", i+1);
            scanf("%d %d", &p[i].at, &p[i].bt);
        }
        if(c==1)
            non_preemptive(p,n);
        else if(c==2)
            preemptive(p,n);
        else
            printf("Invalid entry");
```

```
    return 0;
}
```

Output:

```
Enter the number of processes:4
Enter choice:
1.Non-preemptive
2.Preemptive
1

Enter the arrival time and burst time for process:
Process 1: 0 7
Process 2: 0 3
Process 3: 0 4
Process 4: 0 6
SJF Non Preemptive Scheduling:
Process AT      BT      CT      TAT     WT
P1      0       7       20      20      13
P2      0       3       3       3       0
P3      0       4       7       7       3
P4      0       6       13      13      7

Average TAT: 10.75 units
Average WT: 5.75 units
Process returned 0 (0x0)   execution time : 24.040 s
Press any key to continue.
```

```
Enter the number of processes:4
Enter choice:
1.Non-preemptive
2.Preemptive
2

Enter the arrival time and burst time for process:
Process 1: 0 7
Process 2: 8 3
Process 3: 3 4
Process 4: 5 6
SJF Preemptive Scheduling:
Process AT      BT      CT      TAT     WT
P1      0       7       7       7       0
P2      8       3       14      6       3
P3      3       4       11      8       4
P4      5       6       20      15      9

Average TAT: 9.00 units
Average WT: 4.00 units
Process returned 0 (0x0)   execution time : 22.664 s
Press any key to continue.
```