Fcfs

```
int main() {
int n,at[10],bt[10],wt[10],tat[10],ct[10],sum,i,j,k;
float totaltat=0,totalwt=0;
printf("enter the total number of processes:");
scanf("%d",&n);
printf("\nEnter The Process Arrival Time & Burst Time\n");
 for(i=0;i < n;i++) {
printf("Enter Arrival time of process[%d]:",i+1);
 scanf("%d",&at[i]);
printf("Enter Burst time of process[%d]:",i+1);
scanf("%d",&bt[i]); }

 sum=at[0];
for(j=0;j < n;j++) {
sum=sum+bt[j];
 ct[j]=sum;
 }

 for(k=0;k < n;k++) {
 tat[k]=ct[k]-at[k];
totaltat=totaltat+tat[k];
}

 for(k=0;k < n;k++) {
 wt[k]=tat[k]-bt[k];
totalwt=totalwt+wt[k];
}
printf("\nProcess\tAT\tBT\tCT\tTAT\tWT\n\n\n");
 for(i=0;i < n;i++) {
```

```
 printf("\nP%d\t %d\t %d\t %d\t %d\t %d\t\n",i+1,at[i],bt[i],ct[i],tat[i],wt[i]);
 }
 printf("\nAverage TurnaroundTime:%f\n",totaltat/n);
printf("\nAverage Waiting Time:%f",totalwt/n);
return 0;
 }
```

Srf

```
#include <stdio.h>
int main() {
   int n, p[10], bt[10], wt[10], tat[10], ct[10], sum, i, j, temp;
   float totaltat = 0, totalwt = 0;
   printf("Enter the total number of processes: ");
   scanf("%d", &n);
   printf("\nEnter the Process IDs\n");
   for (i = 0; i < n; i++) {
      scanf("%d", &p[i]);
   }
   printf("\nEnter the burst times of the processes\n");
   for (i = 0; i < n; i++) {
      printf("Enter Burst Time of Process[%d]: ", p[i]);
      scanf("%d", &bt[i]);
   }
   for (i = 0; i < n; i++) {
      for (j = 0; j < n - i - 1; j++) {
         if (bt[j] > bt[j + 1]) {
            temp = bt[j];
            bt[j] = bt[j + 1];
            bt[j + 1] = temp;
```

```
            temp = p[j];
            p[j] = p[j + 1];
            p[j + 1] = temp;
          }
        }
    }
    sum = 0;
    for (j = 0; j < n; j++) {
        sum += bt[j];
        ct[j] = sum;
    }

    for (i = 0; i < n; i++) {
        tat[i] = ct[i];
        totaltat += tat[i];
        wt[i] = tat[i] - bt[i];
        totalwt += wt[i];
    }
    printf("\nProcess\tBT\tTAT\tWT\n");
    for (i = 0; i < n; i++) {
        printf("P%d\t%d\t%d\t%d\n", p[i], bt[i], tat[i], wt[i]);
    }
 printf("\nAverage Turnaround Time: %.2f\n", totaltat / n);
    printf("Average Waiting Time: %.2f\n", totalwt / n);

    return 0;
}
```

Srtf

```
#include <stdio.h>
#include <limits.h>
int main() {
    int n, time = 0, completed = 0, smallest;
    int at[10], bt[10], rbt[10], wt[10] = {0}, tat[10] = {0}, p[10];
    float totalwt = 0, totaltat = 0;
 printf("Enter the total number of processes: ");
    scanf("%d", &n);
 printf("\nEnter Arrival Time and Burst Time for each process:\n");
    for (int i = 0; i < n; i++) {
        printf("P%d Arrival Time: ", i + 1);
        scanf("%d", &at[i]);
        printf("P%d Burst Time: ", i + 1);
        scanf("%d", &bt[i]);
        rbt[i] = bt[i];
        p[i] = i + 1;
    }

    while (completed != n) {
        smallest = -1;
        int min_time = INT_MAX;
        for (int i = 0; i < n; i++) {
            if (at[i] <= time && rbt[i] > 0 && rbt[i] < min_time) {
                min_time = rbt[i];
                smallest = i;
            }
        }

        if (smallest == -1) {
```

```
            time++;
            continue;
        }
        rbt[smallest]--;
        time++;

        if (rbt[smallest] == 0) {
            completed++;
            tat[smallest] = time - at[smallest];
            wt[smallest] = tat[smallest] - bt[smallest];
        }
    }
    for (int i = 0; i < n; i++) {
        totaltat += tat[i];
        totalwt += wt[i];
    }
   printf("\nProcess\tAT\tBT\tTAT\tWT\n");
    for (int i = 0; i < n; i++) {
        printf("P%d\t%d\t%d\t%d\t%d\n", p[i], at[i], bt[i], tat[i], wt[i]);
    }
  printf("\nAverage Turnaround Time: %.2f\n", totaltat / n);
    printf("Average Waiting Time: %.2f\n", totalwt / n);

    return 0;
}
```

FIFO PAGE REPLACE

```
#include<stdio.h>
void main() {
 int i, j, k, f, pf=0, count=0, rs[25], m[10], n;
 printf("\n Enter the length of reference string -- ");
 scanf("%d",&n); printf("\n Enter the reference string -- ");
 for(i=0;i<n;i++) scanf("%d",&rs[i]);
 printf("\n Enter no. of frames -- ");
 scanf("%d",&f);
 for(i=0;i<f;i++) m[i]=-1;
 printf("\n The Page Replacement Process is -- \n");
 for(i=0;i<n;i++)
 {
 for(k=0;k<f;k++)
 {
 if(m[k]==rs[i])
 break;
  }
 if(k==f)
 {
 m[count++]=rs[i];
 pf++;
  }
 for(j=0;j<f;j++)
 printf("\t%d",m[j]);
 if(k==f)
 printf("\tPF No. %d",pf);
 printf("\n");
 if(count==f) count=0; }
 printf("\n The number of Page Faults using FIFO are %d",pf); }
```

LRU

```
#include<stdio.h>
void main() {
   int i, j, k, f, pf = 0, rs[25], m[10], n, least, index, found;
    printf("\n Enter the length of reference string -- ");
   scanf("%d", &n);
   printf("\n Enter the reference string -- ");
   for(i = 0; i < n; i++)
      scanf("%d", &rs[i]);

   printf("\n Enter number of frames -- ");
   scanf("%d", &f);

   for(i = 0; i < f; i++)
      m[i] = -1; // Initialize frames

   printf("\n The Page Replacement Process is -- \n");

   for(i = 0; i < n; i++) {
      found = 0;
      for(k = 0; k < f; k++) {
         if(m[k] == rs[i]) {
            found = 1;
            break;
         }
      }

      if(found == 0) { // Page fault
         pf++;
         int least_used = 0;
         for(j = 1; j < f; j++) {
```

```
            int is_recent = 0;
            for(k = i - 1; k >= 0; k--) {
               if(m[j] == rs[k]) {
                  is_recent = 1;
                  break;
               }
            }
            if(is_recent == 0) {
               least_used = j;
               break;
            }
         }

         m[least_used] = rs[i]; // Replace LRU page
      }
      for(j = 0; j < f; j++)
         printf("\t%d", m[j]);

      if(found == 0)
         printf("\tPF No. %d", pf);

      printf("\n");
   }
   printf("\n The number of Page Faults using LRU are %d", pf);
}
```

Priority

```
#include<stdio.h>
#include<limits.h>
int main() {
    int n, at[10], bt[10], wt[10], tat[10], ct[10], pr[10], rbt[10];
    int completed = 0, time = 0, i, smallest;
    float totaltat = 0, totalwt = 0;
   printf("Enter the total number of processes: ");
    scanf("%d", &n);
  printf("\nEnter Arrival Time, Burst Time, and Priority for each process:\n");
    for (i = 0; i < n; i++) {
        printf("Process[%d] Arrival Time: ", i + 1);
        scanf("%d", &at[i]);
        printf("Process[%d] Burst Time: ", i + 1);
        scanf("%d", &bt[i]);
        printf("Process[%d] Priority: ", i + 1);
        scanf("%d", &pr[i]);
        rbt[i] = bt[i];
    }

    while (completed != n) {
        smallest = -1;
        int min_priority = INT_MAX;
        for (i = 0; i < n; i++) {
            if (at[i] <= time && rbt[i] > 0 && pr[i] < min_priority) {
                min_priority = pr[i];
                smallest = i;
            }
        }

        if (smallest == -1) {
```

```
            time++;
            continue;
        }
        time += rbt[smallest];
        rbt[smallest] = 0;
        completed++;
        ct[smallest] = time;
        tat[smallest] = ct[smallest] - at[smallest];
        wt[smallest] = tat[smallest] - bt[smallest];
    }
    for (i = 0; i < n; i++) {
        totaltat += tat[i];
        totalwt += wt[i];
    }
    printf("\nProcess\tAT\tBT\tPr\tCT\tTAT\tWT\n");
    for (i = 0; i < n; i++) {
        printf("P%d\t%d\t%d\t%d\t%d\t%d\t%d\n", i + 1, at[i], bt[i], pr[i], ct[i], tat[i], wt[i]);
    }
printf("\nAverage Turnaround Time: %.2f\n", totaltat / n);
    printf("Average Waiting Time: %.2f\n", totalwt / n);
   return 0;
}
```

Round Robin

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

int main() {
   int n, at[10], bt[10], wt[10], tat[10], ct[10], rbt[10];
   int completed = 0, time = 0, i, quantum;
   float totaltat = 0, totalwt = 0;

   printf("Enter the total number of processes: ");
   scanf("%d", &n);

   printf("\nEnter Arrival Time and Burst Time for each process:\n");
   for (i = 0; i < n; i++) {
      printf("Process[%d] Arrival Time: ", i + 1);
      scanf("%d", &at[i]);
      printf("Process[%d] Burst Time: ", i + 1);
      scanf("%d", &bt[i]);
      rbt[i] = bt[i];
   }

   printf("Enter the time quantum: ");
   scanf("%d", &quantum);

   while (completed != n) {
      int all_done = 1;
      for (i = 0; i < n; i++) {
         if (rbt[i] > 0 && at[i] <= time) {
            all_done = 0;
            if (rbt[i] > quantum) {
               time += quantum;
```

```
                rbt[i] -= quantum;
            } else {
                time += rbt[i];
                wt[i] = time - at[i] - bt[i];
                tat[i] = time - at[i];
                rbt[i] = 0;
                completed++;
            }
        }
    }
    if (all_done == 1) {
        time++;
    }
  }

  for (i = 0; i < n; i++) {
    totaltat += tat[i];
    totalwt += wt[i];
  }
+
  printf("\nProcess\tAT\tBT\tCT\tTAT\tWT\n");
  for (i = 0; i < n; i++) {
    ct[i] = tat[i] + at[i];
    printf("P%d\t%d\t%d\t%d\t%d\t%d\n", i + 1, at[i], bt[i], ct[i], tat[i], wt[i]);
  }

  printf("\nAverage Turnaround Time: %.2f\n", totaltat / n);
  printf("Average Waiting Time: %.2f\n", totalwt / n);

  return 0;
}
```