

b. Priority Scheduling

```
#include <stdio.h>

#define MAX_PROC 10

int main() {
    int numProcesses, i, j, clock = 0, processesCompleted = 0;
    int arrivalTime[MAX_PROC], burstTime[MAX_PROC], procPriority[MAX_PROC];
    int waitTime[MAX_PROC], turnAroundTime[MAX_PROC], isDone[MAX_PROC] = {0};

    printf("Enter the number of processes (max %d): ", MAX_PROC);
    scanf("%d", &numProcesses);

    printf("Enter the arrival time, burst time, and priority for each process:\n");
    for(i = 0; i < numProcesses; i++)
        scanf("%d %d %d", &arrivalTime[i], &burstTime[i], &procPriority[i]);

    while(processesCompleted < numProcesses) {
        int selectedProc = -1, maxPriority = -1;
        for(i = 0; i < numProcesses; i++) {
            if(arrivalTime[i] <= clock && !isDone[i] && procPriority[i] > maxPriority) {
                maxPriority = procPriority[i];
                selectedProc = i;
            }
        }
        if(selectedProc != -1) {
            waitTime[selectedProc] = clock - arrivalTime[selectedProc];
            clock += burstTime[selectedProc];
            turnAroundTime[selectedProc] = clock - arrivalTime[selectedProc];
            isDone[selectedProc] = 1;
            processesCompleted++;
        } else {
            clock++;
        }
    }

    printf("Proc\tAT\tBT\tPri\tWT\tTAT\n");
    for(i = 0; i < numProcesses; i++)
        printf("%d\t%d\t%d\t%d\t%d\t%d\n", i+1, arrivalTime[i], burstTime[i], procPriority[i], waitTime[i], turnAroundTime[i]);
}
```

```
return 0;  
}
```

Enter the number of processes (max 10): 4

Enter the arrival time, burst time, and priority for each process:

0 3 2

1 5 3

2 1 4

3 2 1

Proc	AT	BT	Pri	WT	TAT
1	0	3	2	0	3
2	1	5	3	2	7
3	2	1	4	5	6
4	3	2	1	6	8