main.c

Share Run

```c
#include <stdio.h>
int main() {
    int n, i, j;
    printf("Enter the number of processes: ");
    scanf("%d", &n);
    int burst_time[n], priority[n], process[n], completion_time[n], turnaround_time[n], waiting_time[n];
    printf("Enter Burst Times and Priorities for each process:\n");
    for (i = 0; i < n; i++) {
        printf("Process %d Burst Time: ", i + 1);
        scanf("%d", &burst_time[i]);
        printf("Process %d Priority: ", i + 1);
        scanf("%d", &priority[i]);
        process[i] = i + 1;
    }
    for (i = 0; i < n - 1; i++) {
        for (j = 0; j < n - i - 1; j++) {
            if (priority[j] > priority[j + 1]) {
                int temp = priority[j];
                priority[j] = priority[j + 1];
                priority[j + 1] = temp;
                temp = burst_time[j];
                burst_time[j] = burst_time[j + 1];
                burst_time[j + 1] = temp;
                temp = process[j];
                process[j] = process[j + 1];
                process[j + 1] = temp;
            }
        }
    }
    completion_time[0] = burst_time[0];
    for (i = 1; i < n; i++) {
        completion_time[i] = completion_time[i - 1] + burst_time[i];
    }
    for (i = 0; i < n; i++) {
        turnaround_time[i] = completion_time[i];
        waiting_time[i] = turnaround_time[i] - burst_time[i];
    }
    printf("\nProcess\tPriority\tBurst Time\tCompletion Time\tTurnaround Time\tWaiting Time\n");
    for (i = 0; i < n; i++) {
        printf("%d\t%d\t\t%d\t\t%d\t\t%d\t\t%d\n", process[i], priority[i], burst_time[i],
            completion_time[i], turnaround_time[i], waiting_time[i]);
    }

    return 0;
}
```

Output

```
Enter the number of processes: 2
Enter Burst Times and Priorities for each process:
Process 1 Burst Time: 5
Process 1 Priority: 2
Process 2 Burst Time: 3
Process 2 Priority: 1

Process Priority    Burst Time  Completion Time Turnaround Time Waiting Time
2   1       3       3       3       0
1   2       5       8       8       3


=== Code Execution Successful ===
```

main.c

Share Run

```c
#include <fcntl.h>
#include <unistd.h>
#include <stdlib.h>
#define BUFFER_SIZE 1024
int main(int argc, char *argv[]) {
    if (argc != 3) {
        fprintf(stderr, "Usage: %s <source_file> <destination_file>\n", argv[0]);
        exit(1);
    }
    int src_fd = open(argv[1], O_RDONLY);
    if (src_fd < 0) {
        perror("Error opening source file");
        exit(1);
    }
    int dest_fd = open(argv[2], O_WRONLY | O_CREAT | O_TRUNC, 0644);
    if (dest_fd < 0) {
        perror("Error opening/creating destination file");
        close(src_fd);
        exit(1);
    }
    char buffer[BUFFER_SIZE];
    ssize_t bytes_read, bytes_written;
    while ((bytes_read = read(src_fd, buffer, BUFFER_SIZE)) > 0) {
        bytes_written = write(dest_fd, buffer, bytes_read);
        if (bytes_written != bytes_read) {
            perror("Error writing to destination file");
            close(src_fd);
            close(dest_fd);
            exit(1);
        }
    }
    if (bytes_read < 0) {
        perror("Error reading source file");
    }
    close(src_fd);
    close(dest_fd);
    printf("File copied successfully.\n");
    return 0;
}
```

Output

```
Usage: /tmp/hf8wJDPmO9/main.o <source_file> <destination_file>


=== Code Exited With Errors ===
```