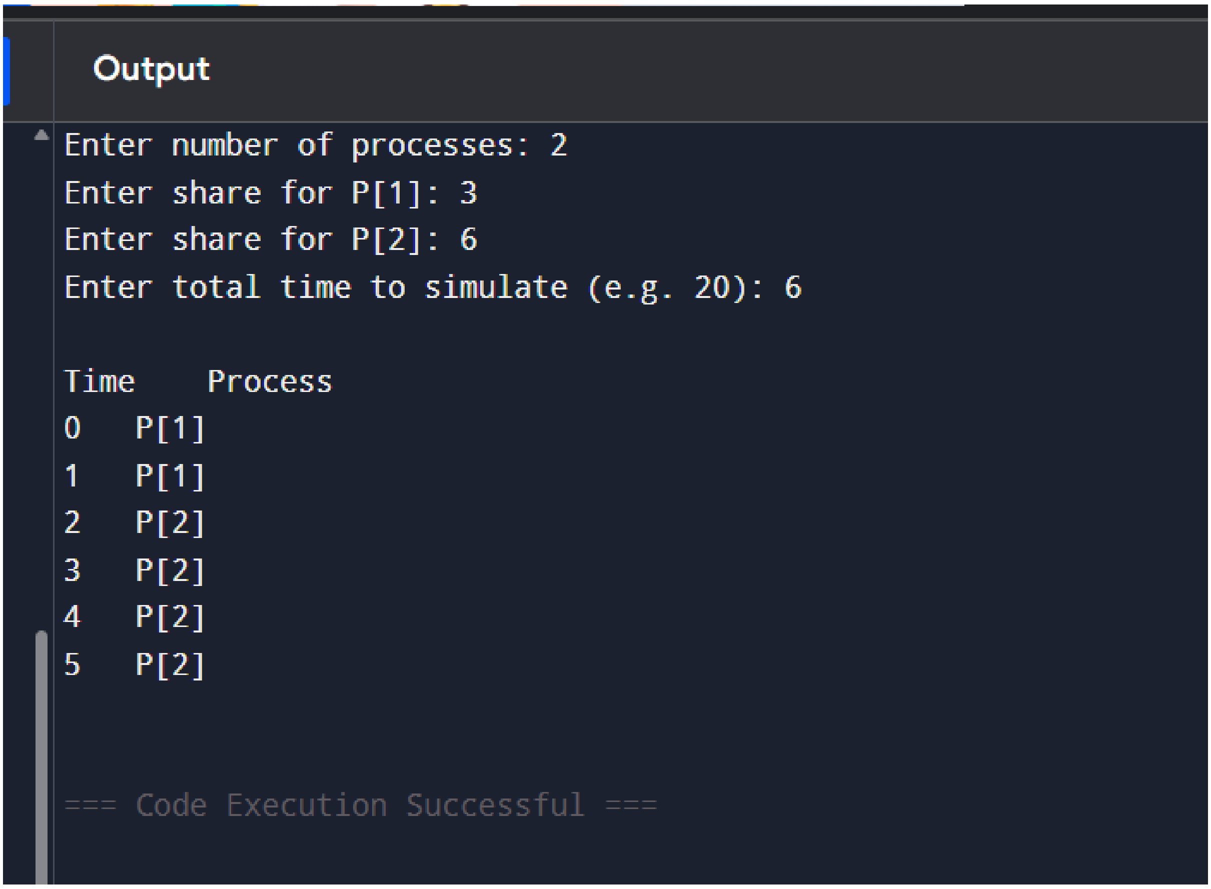
c) Proportional scheduling

```
#include <stdio.h>

struct Process {
   int id;
   int share;
   int executed;
};

int main() {
   int n, total_time;
   struct Process p[10];

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

   int total_share = 0;
   for(int i = 0; i < n; i++) {
      printf("Enter share for P[%d]: ", i+1);
      scanf("%d", &p[i].share);
      p[i].id = i + 1;
      p[i].executed = 0;
      total_share += p[i].share;
   }

   printf("Enter total time to simulate (e.g. 20): ");
   scanf("%d", &total_time);

   printf("\nTime\tProcess\n");

   int time = 0, quota[10] = {0};
   while(time < total_time) {
      for(int i = 0; i < n && time < total_time; i++) {
         int run_for = (p[i].share * total_time) / total_share;
         while(p[i].executed < run_for && time < total_time) {
            printf("%d\tP[%d]\n", time, p[i].id);
            p[i].executed++;
            time++;
         }
      }
   }

   return 0;
}
```



```
Enter number of processes: 2
Enter share for P[1]: 3
Enter share for P[2]: 6
Enter total time to simulate (e.g. 20): 6

Time    Process
0   P[1]
1   P[1]
2   P[2]
3   P[2]
4   P[2]
5   P[2]


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