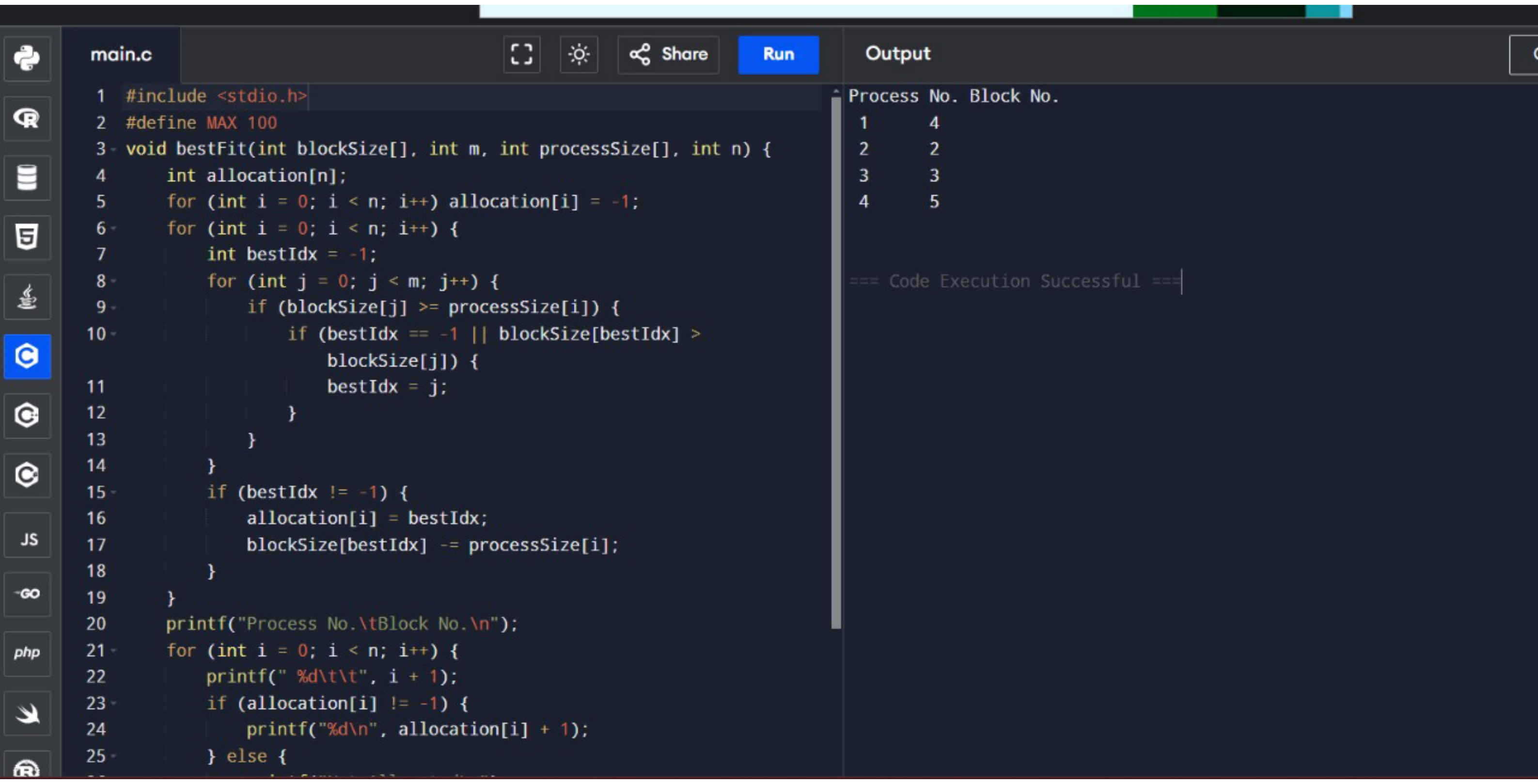
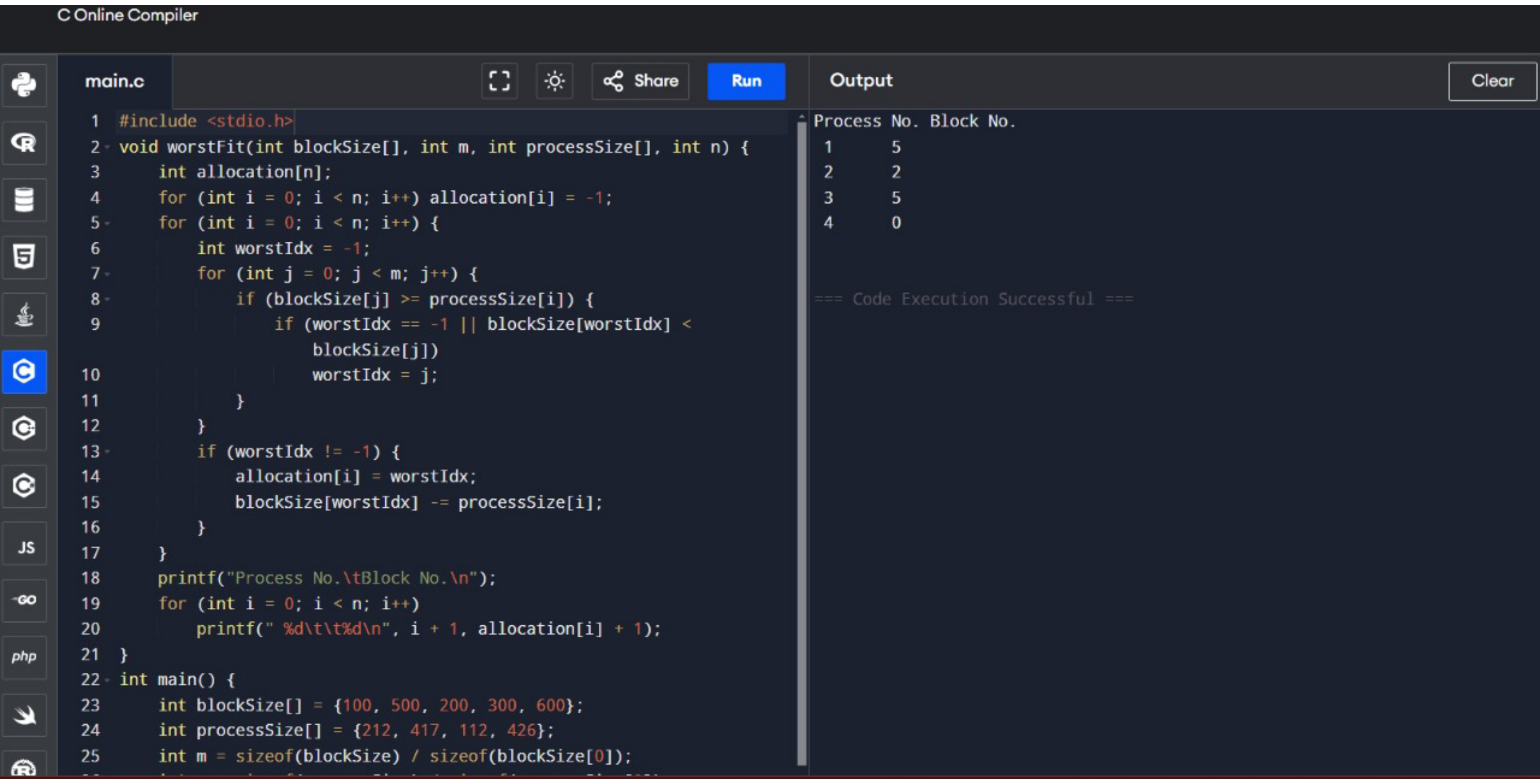


```c
#include <stdio.h>
#define MAX 100
void bestFit(int blockSize[], int m, int processSize[], int n) {
    int allocation[n];
    for (int i = 0; i < n; i++) allocation[i] = -1;
    for (int i = 0; i < n; i++) {
        int bestIdx = -1;
        for (int j = 0; j < m; j++) {
            if (blockSize[j] >= processSize[i]) {
                if (bestIdx == -1 || blockSize[bestIdx] > blockSize[j]) {
                    bestIdx = j;
                }
            }
        }
        if (bestIdx != -1) {
            allocation[i] = bestIdx;
            blockSize[bestIdx] -= processSize[i];
        }
    }
    printf("Process No.\tBlock No.\n");
    for (int i = 0; i < n; i++) {
        printf(" %d\t\t", i + 1);
        if (allocation[i] != -1) {
            printf("%d\n", allocation[i] + 1);
        } else {
```

```
Process No. Block No.
 1       4
 2       2
 3       3
 4       5


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



```c
#include <stdio.h>
void worstFit(int blockSize[], int m, int processSize[], int n) {
    int allocation[n];
    for (int i = 0; i < n; i++) allocation[i] = -1;
    for (int i = 0; i < n; i++) {
        int worstIdx = -1;
        for (int j = 0; j < m; j++) {
            if (blockSize[j] >= processSize[i]) {
                if (worstIdx == -1 || blockSize[worstIdx] < blockSize[j])
                    worstIdx = j;
            }
        }
        if (worstIdx != -1) {
            allocation[i] = worstIdx;
            blockSize[worstIdx] -= processSize[i];
        }
    }
    printf("Process No.\tBlock No.\n");
    for (int i = 0; i < n; i++)
        printf(" %d\t\t%d\n", i + 1, allocation[i] + 1);
}
int main() {
    int blockSize[] = {100, 500, 200, 300, 600};
    int processSize[] = {212, 417, 112, 426};
    int m = sizeof(blockSize) / sizeof(blockSize[0]);
```

```
Process No. Block No.
 1       5
 2       2
 3       5
 4       0


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