5.Assignment Problem

```
#include <stdio.h>

#define N 3

int cost[N][N], mask[N][N], rowCover[N], colCover[N];

void subtractRowMinima() {
   for (int i = 0; i < N; i++) {
      int min = cost[i][0];
      for (int j = 1; j < N; j++)
         if (cost[i][j] < min)
            min = cost[i][j];
      for (int j = 0; j < N; j++)
         cost[i][j] -= min;
   }
}

void subtractColMinima() {
   for (int j = 0; j < N; j++) {
      int min = cost[0][j];
      for (int i = 1; i < N; i++)
         if (cost[i][j] < min)
            min = cost[i][j];
      for (int i = 0; i < N; i++)
         cost[i][j] -= min;
   }
}

void coverZeros() {
   for (int i = 0; i < N; i++)
```

```
        rowCover[i] = 0;
    for (int j = 0; j < N; j++)
        colCover[j] = 0;

    for (int i = 0; i < N; i++)
        for (int j = 0; j < N; j++)
            if (cost[i][j] == 0 && !rowCover[i] && !colCover[j]) {
                mask[i][j] = 1;
                rowCover[i] = 1;
                colCover[j] = 1;
            }

    for (int i = 0; i < N; i++)
        rowCover[i] = 0;
    for (int j = 0; j < N; j++)
        colCover[j] = 0;
}

void printAssignment() {
    int total = 0;
    printf("Assignments:\n");
    for (int i = 0; i < N; i++)
        for (int j = 0; j < N; j++)
            if (mask[i][j]) {
                printf("Worker %d assigned to Job %d with cost %d\n", i + 1, j + 1, cost[i][j]);
                total += cost[i][j];
            }
    printf("Total cost: %d\n", total);
}

int main() {
```

```
    printf("Enter the cost matrix (%dx%d):\n", N, N);
    for (int i = 0; i < N; i++)
        for (int j = 0; j < N; j++)
            scanf("%d", &cost[i][j]);

    subtractRowMinima();
    subtractColMinima();
    coverZeros();
    printAssignment();

    return 0;
}
```

Output

```
Enter the cost matrix (3x3):
3
3
6
9
3
36
5
3
3
Assignments:
Worker 1 assigned to Job 1 with cost 0
Worker 2 assigned to Job 2 with cost 0
Worker 3 assigned to Job 3 with cost 0
Total cost: 0


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