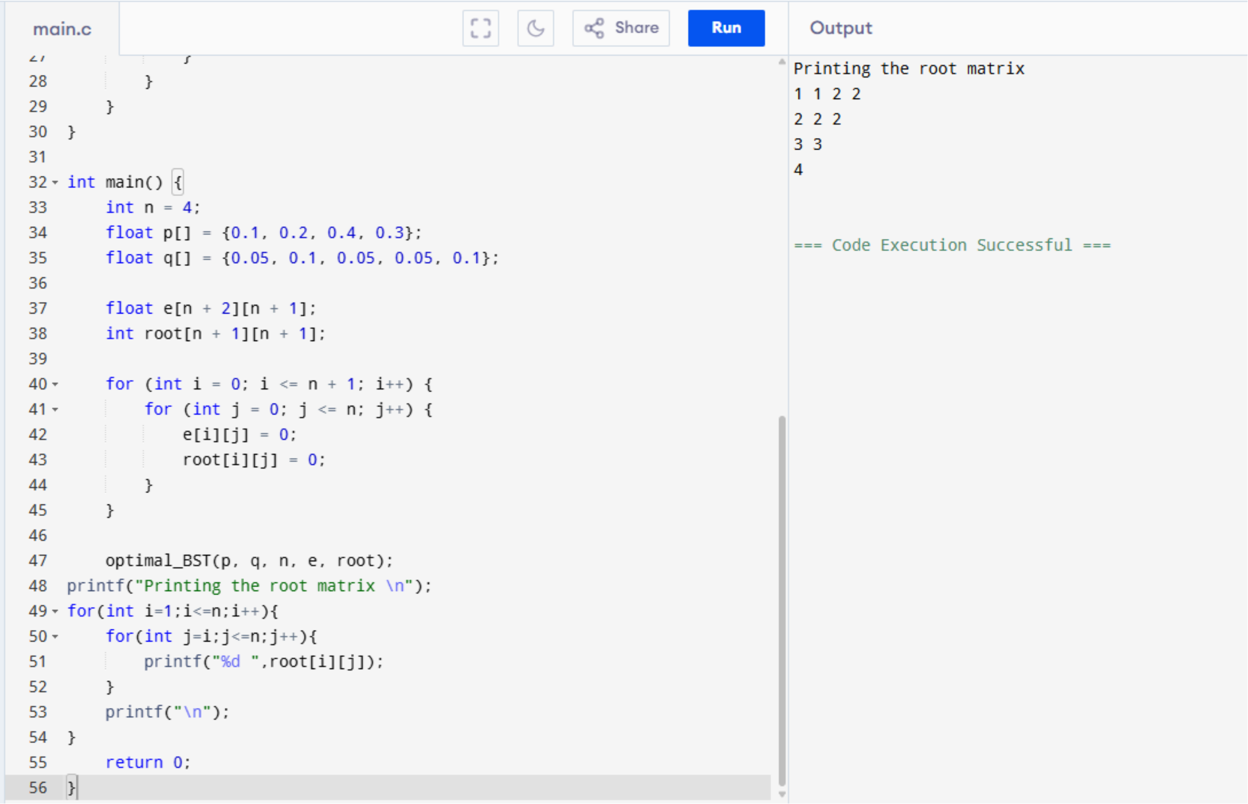
Pract6
A8_B2_18

```
#include <stdio.h>
#include <float.h>

#define INFINITY FLT_MAX

void optimal_BST(float p[], float q[], int n, float e[][n+1], int root[][n+1]) {
    float w[n+1][n+1];
    int i, j, l, r, t;

    for (i = 1; i <= n + 1; i++) {
        e[i][i - 1] = q[i - 1];
        w[i][i - 1] = q[i - 1];
    }

    for (l = 1; l <= n; l++) {
        for (i = 1; i <= n - l + 1; i++) {
            j = i + l - 1;
            e[i][j] = INFINITY;
            w[i][j] = w[i][j - 1] + p[j - 1] + q[j];

            for (r = i; r <= j; r++) {
                t = e[i][r - 1] + e[r + 1][j] + w[i][j];
                if (t < e[i][j]) {
                    e[i][j] = t;
                    root[i][j] = r;
                }
            }
        }
    }
}

int main() {
    int n = 4;
    float p[] = {0.1, 0.2, 0.4, 0.3};
    float q[] = {0.05, 0.1, 0.05, 0.05, 0.1};

    float e[n + 2][n + 1];
    int root[n + 1][n + 1];

    for (int i = 0; i <= n + 1; i++) {
        for (int j = 0; j <= n; j++) {
            e[i][j] = 0;
            root[i][j] = 0;
        }
    }

    optimal_BST(p, q, n, e, root);
printf("Printing the root matrix \n");
for(int i=1;i<=n;i++){
    for(int j=i;j<=n;j++){
        printf("%d ",root[i][j]);
    }
    printf("\n");
}

    return 0;
}
```



```
            }
        }
    }
}

int main() {
    int n = 4;
    float p[] = {0.1, 0.2, 0.4, 0.3};
    float q[] = {0.05, 0.1, 0.05, 0.05, 0.1};

    float e[n + 2][n + 1];
    int root[n + 1][n + 1];

    for (int i = 0; i <= n + 1; i++) {
        for (int j = 0; j <= n; j++) {
            e[i][j] = 0;
            root[i][j] = 0;
        }
    }

    optimal_BST(p, q, n, e, root);
printf("Printing the root matrix \n");
for(int i=1;i<=n;i++){
    for(int j=i;j<=n;j++){
        printf("%d ",root[i][j]);
    }
    printf("\n");
}
    return 0;
}
```

```
Printing the root matrix
1 1 2 2
2 2 2
3 3
4


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