**#include <iostream>**

**using namespace std;**

**const int n = 4;**

**const int MAX = 1000000;**

**int dist[n + 1][n + 1] = {**

**{ 0, 0, 0, 0, 0 }, { 0, 0, 10, 15, 20 },**

**{ 0, 10, 0, 25, 25 }, { 0, 15, 25, 0, 30 },**

**{ 0, 20, 25, 30, 0 },**

**};**

**int memo[n + 1][1 << (n + 1)];**

**int fun(int i, int mask)**

**{**

**if (mask == ((1 << i) | 3))**

**return dist[1][i];**

**// memoization**

**if (memo[i][mask] != 0)**

**return memo[i][mask];**

**int res = MAX; // result of this sub-problem**

**for (int j = 1; j <= n; j++)**

**if ((mask & (1 << j)) && j != i && j != 1)**

**res = std::min(res, fun(j, mask & (~(1 << i)))**

**+ dist[j][i]);**

**return memo[i][mask] = res;**

**}**

**// Driver program to test above logic**

**int main()**

**{**

**int ans = MAX;**

**for (int i = 1; i <= n; i++)**

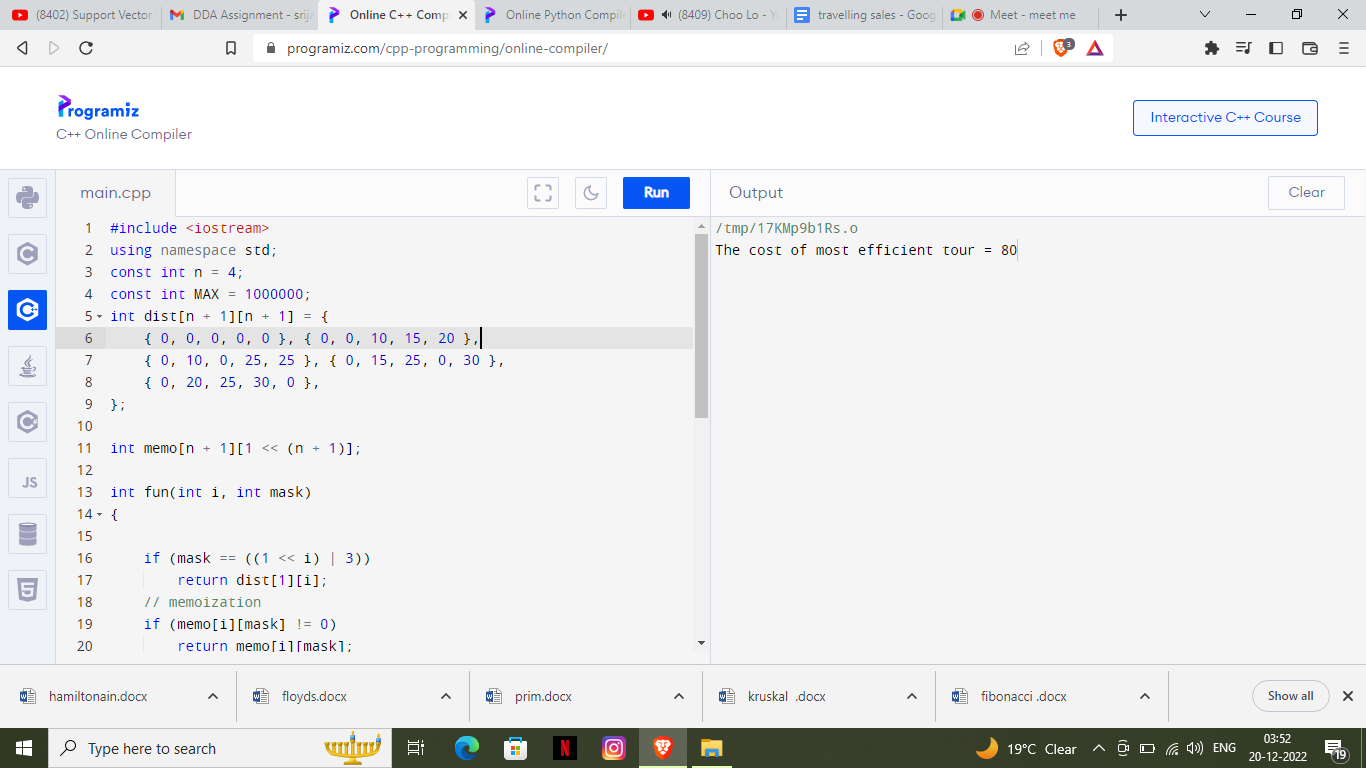
**ans = std::min(ans, fun(i, (1 << (n + 1)) - 1)**

**+ dist[i][1]);**

**printf("The cost of most efficient tour = %d", ans);**

**return 0;**

**}**

****