{

    "": {

        "prefix": "prefix\_sum\_2d\_array",

        "body": [

          "const int pre\_limit\_2d=1e3+10;",

          "ll a[pre\_limit\_2d][pre\_limit\_2d];",

          "ll pref[pre\_limit\_2d][pre\_limit\_2d];",

          "",

          "",

          "void prefix\_sum\_2d\_array()",

          "{",

          "    int n;",

          "    cin>>n;",

          "",

          "    //taking input for prefix sum for 2d array for n\*n square matrix; it can be rectangular as well.",

          "    for(int i=1; i<=n; i++) //in prefix sum we need to initialize i with 1; not 0.",

          "    {",

          "        for(int j=1; j<=n; j++)",

          "        {",

          "            cin>>a[i][j];",

          "            pref[i][j] = a[i][j] + pref[i-1][j] + pref[i][j-1] - pref[i-1][j-1];",

          "        }",

          "    }",

          "",

          "    //for q queries we will print the prefix sum form a point to another point (diagonal points will be given)",

          "    /\*",

          "        ............",

          "        ......sXXX..",

          "        ......XXXX..",

          "        ......XXXe..",

          "",

          "        here s and e are 2 points of the diagonal and we are going to calculate the sum of ",

          "        are rectangle staring from s and ending at e. ",

          "    \*/",

          "",

          "    int q;",

          "    cin>>q;",

          "",

          "    while(q--)",

          "    {",

          "        int x1, y1, x2, y2;",

          "        cin>>x1>>y1>>x2>>y2;",

          "        cout<<pref[x2][y2] - pref[x1-1][y2] - pref[x2][y1-1] + pref[x1-1][y1-1]<<endl;",

          "    }",

          "}"

        ],

        "description": ""

      }

}