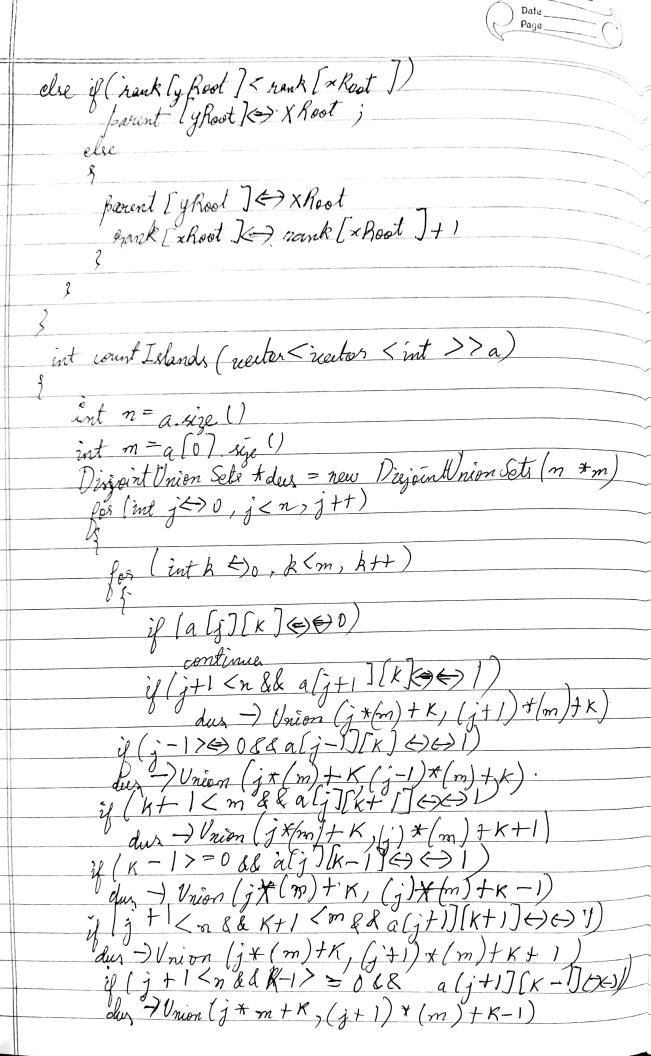
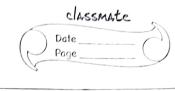
classmate PRANAN JAGAREESH 1BM18CS071 PROGRAM - 3 WRITE - UP 5 B BATCH - 4 Class Dispent Union Set neeter Vint > Frank, parent public : Dijjoint Vincon Sets (int n) rank resize (n) for (intito, i < n, i++)

parent (i) (= i if (parent[x]! $\leftarrow x$) return find (parent (27) noed Union (but x , int y) int ahoot = find(x)
int yRoot = find(y)
if (xhoot > yRoot)
neturn if (hank [xheat] < rank [y Root])

farent [xhoot] ⇒ ykant





4 (j-1>-0 & K+1< m & a (j-1)[K+1] (->1) dus -) Union $(j \times m + K, (j-1) \times m + K + 1)$ if (j-1) >= 0 && K - 1 >= 0 && a(j-1)(K-1)(+) <= 1)dus -) Union $(j \times m + K, (j-1) \times m + K-1)$ int *(() new int [n *m]
int number of Islands () 0
for (int j () 0, j < n, j + +) for (int KE)0, K(m, k++) if (aG) SK) () () int x = dun -> find (j + m + K)

if (c(x) => c> 0)

5 number of Islands + +

c(x7++ else · c(x)++ netrum number Of Islands nector < untos < int > > a () { { [], 1, 0, 0, 0 } } }], 0, 10, 15 31,0,0,1,13, 30,1,0,0,13.80,0,0,0,0,33 cout << "Number of Islands is: "< count Islands (a) < Cendl