Bai1:

a.

Sau đây là một vì dụ bằng python để chứng minh :

**Trước khi tối ưu lệnh lặp và một số biến phụ:**

def sum\_of\_all\_pairs(arr):

n = len(arr)

total\_sum = 0

for i in range(n):

for j in range(n):

total\_sum += arr[i] + arr[j]

return total\_sum

arr = [1, 2, 3, 4]

print(sum\_of\_all\_pairs(arr)) # Output: 100

**Sau khi tối ưu lệnh lặp và một số biến phụ:**

def optimized\_sum\_of\_all\_pairs(arr):

total\_sum = 0

for num in arr:

total\_sum += num

return 2 \* total\_sum \* len(arr)

arr = [1, 2, 3, 4]

print(optimized\_sum\_of\_all\_pairs(arr)) # Output: 100

* Bằng cách giảm thiểu tối đa sự trùng lặp hoặc dư thừa không cần thiết khi lặp sẽ giảm được số vòng lặp .Như bài toán trên để tìm được tổng từng cặp cạnh nhau bằng cách gấp đôi tổng của cả mảng sẽ cho ra cùng đáp án cần tìm.

b.

int F( int X[] , int n, int m, int k) {

is\_one = (X[0] == 1);

count= 0;

qualify = 0;

for ( int i = 0; i < n ; i++) {

if ( X[i] == is\_one){

count ++;

if( (is\_one && count == k) || (!is\_one && count == m)){

qualify ++;

} else if ( (is\_one && count > k) || (!is\_one && count > m)){

return 0;

}

} else {

count = 1;

is\_one = !is\_one ;

}

}

return qualify == 2;

}

c.

* + X[] = {1, 0, 0, 0, 1, 1, 1}. 🡺 F() = 0
  + X[] = {1, 0, 0, 1, 1, 1, 1}. 🡺 F() = 0
  + X[] = {1, 0, 0, 1, 1, 1, 0}.- 🡺F() = 1