Reverse Part of Array

Java Code:

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    public static void main(String[] args) {
         Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int[] arr = new int[n];
        for(int i=0;i<n;i++){
            arr[i] = scn.nextInt();
        }
        int s = scn.nextInt();
        int e = scn.nextInt();

        reversePart(arr,s,e);

        for(int i=0;i<n;i++){
            System.out.print(arr[i]+" ");
        }
    }

    public static void reversePart(int[]arr, int s, int e){
        int sp = s;
        int ep = e;
        while(sp < ep){
            int temp = arr[sp];
            arr[sp] = arr[ep];
            arr[ep] = temp;
            sp++;
            ep--;
        }

    }
}
```

C++ Code:

```
#include <iostream>
#include <vector>
using namespace std;

void reversePart(vector<int>& arr, int s, int e) {
    int sp = s;
    int ep = e;
    while (sp < ep) {
        int temp = arr[sp];
        arr[sp] = arr[ep];
        arr[ep] = temp;
        sp++;
        ep--;
    }
}

int main() {
    int n;
    cin >> n;
    vector<int> arr(n);
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }
    int s, e;
    cin >> s >> e;
    reversePart(arr, s, e);
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }
    return 0;
}
```

Python Code:

```
def reverse_part(arr, s, e):
    sp = s
    ep = e
    while sp < ep:
        arr[sp], arr[ep] = arr[ep], arr[sp]
        sp += 1
        ep -= 1
```

```
def main():
    n = int(input())
    arr = list(map(int, input().split()))

    s, e = map(int, input().split())

    reverse_part(arr, s, e)

    for num in arr:
        print(num, end=" ")

if __name__ == "__main__":
    main()
```

Rotate Array

Java Code:

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    public static void main(String[] args) {
         Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int[] arr = new int[n];
        for(int i=0;i<n;i++){
            arr[i] = scn.nextInt();
        }
        int k = scn.nextInt();

        k = k%n;
        reversePart(arr,0,n-1);
        reversePart(arr,0,k-1);
        reversePart(arr,k,n-1);

        for(int i=0;i<n;i++){
            System.out.print(arr[i]+" ");
        }
    }

    public static void reversePart(int[]arr, int s, int e){
        int sp = s;
        int ep = e;
        while(sp < ep){
            int temp = arr[sp];
            arr[sp] = arr[ep];
            arr[ep] = temp;
            sp++;
            ep--;
        }

    }
}
```

C++ Code:

```
#include <iostream>
#include <vector>
using namespace std;

void reversePart(vector<int>& arr, int s, int e) {
    int sp = s;
    int ep = e;
    while (sp < ep) {
        int temp = arr[sp];
        arr[sp] = arr[ep];
        arr[ep] = temp;
        sp++;
        ep--;
    }
}

int main() {
    int n;
    cin >> n;
    vector<int> arr(n);
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }
    int k;
    cin >> k;
    k = k % n;
    reversePart(arr, 0, n - 1);
    reversePart(arr, 0, k - 1);
    reversePart(arr, k, n - 1);
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }
    return 0;
}
```

Python Code:

```
def reverse_part(arr, s, e):
    sp = s
    ep = e
    while sp < ep:
        arr[sp], arr[ep] = arr[ep], arr[sp]
        sp += 1
        ep -= 1

def main():
    n = int(input())
    arr = list(map(int, input().split()))

    k = int(input())
    k = k % n

    reverse_part(arr, 0, n - 1)
    reverse_part(arr, 0, k - 1)
    reverse_part(arr, k, n - 1)

    for num in arr:
        print(num, end=" ")

if __name__ == "__main__":
    main()
```

Count Greater

Java Code:

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    public static void main(String[] args) {
         Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int[] arr = new int[n];
        for(int i=0;i<n;i++){
            arr[i] = scn.nextInt();
        }

        int max = Integer.MIN_VALUE;
        for(int i=0;i<n;i++){
            if(arr[i] > max){
                max = arr[i];
            }
        }

        int count = 0;
        for(int i=0;i<n;i++){
            if(arr[i] == max){
                count++;
            }
        }
        System.out.println(n - count);
    }
}
```

C++ Code:

```
#include <iostream>
using namespace std;

int main() {
    int n;
    cin >> n;

    int* arr = new int[n];
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }

    int max = INT_MIN;
    for (int i = 0; i < n; i++) {
        if (arr[i] > max) {
            max = arr[i];
        }
    }

    int count = 0;
    for (int i = 0; i < n; i++) {
        if (arr[i] == max) {
            count++;
        }
    }

    cout << n - count << endl;

    delete[] arr;
    return 0;
}
```

Python Code:

```
def main():
    n = int(input())

    arr = list(map(int, input().split()))

    max_val = float('-inf')
    for num in arr:
        if num > max_val:
            max_val = num

    count = 0
    for num in arr:
        if num == max_val:
            count += 1

    print(n - count)

if __name__ == "__main__":
    main()
```

Two Sum Brute

Java Code:

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int[]arr = new int[n];

        for(int i=0;i<n;i++){
            arr[i] = scn.nextInt();
        }

        int k = scn.nextInt();

        System.out.println(twosum(arr,k));



    }

    public static boolean twosum(int[] arr, int k){
        int n = arr.length;
        for(int i=0;i<n-1;i++){
            for(int j = i+1;j<n;j++){
                if(arr[i] + arr[j] == k){
                    return true;
                }
            }
        }

        return false;
    }
}
```

C++ Code:

```cpp
#include <iostream>
#include <vector>
using namespace std;

bool two_sum(const vector<int>& arr, int k) {
    int n = arr.size();
    for (int i = 0; i < n - 1; i++) {
        for (int j = i + 1; j < n; j++) {
            if (arr[i] + arr[j] == k) {
                return true;
            }
        }
    }
    return false;
}

int main() {
    int n;
    cin >> n;

    vector<int> arr(n);
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }

    int k;
    cin >> k;

    cout << (two_sum(arr, k) ? "true" : "false") << endl;

    return 0;
}
```

Python Code:

```
def two_sum(arr, k):
    n = len(arr)
    for i in range(n - 1):
        for j in range(i + 1, n):
            if arr[i] + arr[j] == k:
                return True
    return False

def main():
    n = int(input())
    arr = list(map(int, input().split()))

    k = int(input())

    print("true" if two_sum(arr, k) else "false")

if __name__ == "__main__":
    main()
```

Max Difference 1

Solution Vid: https://youtu.be/vlbcMdMgY5o

Java Code:

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

// "static void main" must be defined in a public class.
public class Main {
    public static void main(String[] args) {

       Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int[]arr = new int[n];
        for(int i=0;i<n;i++){
            arr[i] = scn.nextInt();
        }

        System.out.println(maxdifference_1(arr));

    }

    public static int maxdifference_1(int[]arr){
        int max = Integer.MIN_VALUE;
        int min = Integer.MAX_VALUE;

        for(int i=0;i<arr.length;i++){
            if(arr[i] > max){
                max = arr[i];
            }

            if(arr[i] < min){
                min = arr[i];
            }

        }

        return max - min;
    }
}
```

C++ Code:

```
#include <iostream>
#include <vector>
using namespace std;

int max_difference_1(const vector<int>& arr) {
    int max_val = INT_MIN;
    int min_val = INT_MAX;

    for (int num : arr) {
        if (num > max_val) {
            max_val = num;
        }

        if (num < min_val) {
            min_val = num;
        }
    }
    return max_val - min_val;
}

int main() {
    int n;
    cin >> n;

    vector<int> arr(n);
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }

    cout << max_difference_1(arr) << endl;

    return 0;
}
```

Python Code:

```
def max_difference_1(arr):
    max_val = float('-inf')
    min_val = float('inf')

    for num in arr:
        if num > max_val:
            max_val = num

        if num < min_val:
            min_val = num

    return max_val - min_val

def main():
    n = int(input())
    arr = list(map(int, input().split()))

    print(max_difference_1(arr))

if __name__ == "__main__":
    main()
```

Max Difference 2

Solution Vid: https://youtu.be/njkBFHgz05Q

Java Code:

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

// "static void main" must be defined in a public class.
public class Main {
    public static void main(String[] args) {

       Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int[]arr = new int[n];

        for(int i=0;i<n;i++){
            arr[i] = scn.nextInt();
        }

        System.out.println(maxdifference_1(arr));

    }

    public static int maxdifference_1(int[]arr){
        for(int i=0;i<arr.length;i++){
            arr[i] = arr[i] + i;
        }

        int max = Integer.MIN_VALUE;
        int min = Integer.MAX_VALUE;

        for(int i=0;i<arr.length;i++){
            if(arr[i] > max){
                max = arr[i];
            }

            if(arr[i] < min){
                min = arr[i];
            }
        }

        return max - min;
    }
}
```

C++ Code:

```
#include <iostream>
#include <vector>
using namespace std;

int max_difference_1(vector<int>& arr) {
    int n = arr.size();

    for (int i = 0; i < n; i++) {
        arr[i] = arr[i] + i;
    }

    int max_val = INT_MIN;
    int min_val = INT_MAX;

    for (int num : arr) {
        if (num > max_val) {
            max_val = num;
        }

        if (num < min_val) {
            min_val = num;
        }
    }

    return max_val - min_val;
}

int main() {
    int n;
    cin >> n;

    vector<int> arr(n);
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }
    cout << max_difference_1(arr) << endl;

    return 0;
}
```

Python Code:

```
def max_difference_1(arr):
    for i in range(len(arr)):
        arr[i] = arr[i] + i

    max_val = float('-inf')
    min_val = float('inf')

    for num in arr:
        if num > max_val:
            max_val = num

        if num < min_val:
            min_val = num

    return max_val - min_val

def main():
    n = int(input())
    arr = list(map(int, input().split()))

    print(max_difference_1(arr))

if __name__ == "__main__":
    main()
```

Max Difference 3

Solution Vid: https://youtu.be/MUFdnVghGkY

Java Code:

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

// "static void main" must be defined in a public class.
public class Main {
    public static void main(String[] args) {

       Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int[]arr = new int[n];

        for(int i=0;i<n;i++){
            arr[i] = scn.nextInt();
        }

        System.out.println(maxdifference_1(arr));

    }

    public static int maxdifference_1(int[]arr){
        for(int i=0;i<arr.length;i++){
            arr[i] = arr[i] - i;
        }

        int max = Integer.MIN_VALUE;
        int min = Integer.MAX_VALUE;

        for(int i=0;i<arr.length;i++){
            if(arr[i] > max){
                max = arr[i];
            }
            if(arr[i] < min){
                min = arr[i];
            }

        }

        return max - min;
    }
}
```

C++ Code:

```
#include <iostream>
#include <vector>
using namespace std;

int max_difference_1(vector<int>& arr) {
    int n = arr.size();

    for (int i = 0; i < n; i++) {
        arr[i] = arr[i] - i;
    }

    int max_val = INT_MIN;
    int min_val = INT_MAX;

    for (int num : arr) {
        if (num > max_val) {
            max_val = num;
        }

        if (num < min_val) {
            min_val = num;
        }
    }

    return max_val - min_val;
}

int main() {
    int n;
    cin >> n;

    vector<int> arr(n);
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }

    cout << max_difference_1(arr) << endl;

    return 0;
}
```

Python Code:

```
def max_difference_1(arr):
    for i in range(len(arr)):
        arr[i] = arr[i] - i

    max_val = float('-inf')
    min_val = float('inf')

    for num in arr:
        if num > max_val:
            max_val = num

        if num < min_val:
            min_val = num

    return max_val - min_val

def main():
    n = int(input())
    arr = list(map(int, input().split()))

    print(max_difference_1(arr))

if __name__ == "__main__":
    main()
```

Max Difference 4

Solution Vid: https://youtu.be/SnzF5MAaO3w

Java Code:

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

// "static void main" must be defined in a public class.
public class Main {
    public static void main(String[] args) {

       Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int[]arr = new int[n];

        for(int i=0;i<n;i++){
            arr[i] = scn.nextInt();
        }

        int ans1 = maxdifference_2(arr);
        int ans2 = maxdifference_3(arr);

        System.out.println(Math.max(ans1,ans2));

    }

     public static int maxdifference_3(int[]arr){
        for(int i=0;i<arr.length;i++){
            arr[i] = arr[i] - i;
        }

        int max = Integer.MIN_VALUE;
        int min = Integer.MAX_VALUE;

        for(int i=0;i<arr.length;i++){
            if(arr[i] > max){
                max = arr[i];
            }

            if(arr[i] < min){
                min = arr[i];
            }

        }

         for(int i=0;i<arr.length;i++){
            arr[i] = arr[i] + i;
        }
        return max - min;
    }


    public static int maxdifference_2(int[]arr){
        for(int i=0;i<arr.length;i++){
            arr[i] = arr[i] + i;
        }

        int max = Integer.MIN_VALUE;
        int min = Integer.MAX_VALUE;

        for(int i=0;i<arr.length;i++){
            if(arr[i] > max){
                max = arr[i];
            }

            if(arr[i] < min){
                min = arr[i];
            }

        }
         for(int i=0;i<arr.length;i++){
            arr[i] = arr[i] - i;
        }

        return max - min;
    }
}
```

C++ Code:

```
#include <iostream>
#include <vector>
using namespace std;

int max_difference_2(vector<int>& arr) {
    for (int i = 0; i < arr.size(); i++) {
        arr[i] = arr[i] + i;
    }

    int max_val = INT_MIN;
    int min_val = INT_MAX;

    for (int num : arr) {
        if (num > max_val) {
            max_val = num;
        }

        if (num < min_val) {
            min_val = num;
        }
    }

    for (int i = 0; i < arr.size(); i++) {
        arr[i] = arr[i] - i;
    }

    return max_val - min_val;
}

int max_difference_3(vector<int>& arr) {
    for (int i = 0; i < arr.size(); i++) {
        arr[i] = arr[i] - i;
    }

    int max_val = INT_MIN;
    int min_val = INT_MAX;

    for (int num : arr) {
        if (num > max_val) {
            max_val = num;
        }

        if (num < min_val) {
            min_val = num;
        }
    }

    for (int i = 0; i < arr.size(); i++) {
        arr[i] = arr[i] + i;
    }

    return max_val - min_val;
}
```

```
int main() {
    int n;
    cin >> n;

    vector<int> arr(n);
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }

    int max_val = INT_MIN;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            if (abs(arr[i] - arr[j]) + i - j > max_val) {
                max_val = abs(arr[i] - arr[j]) + i - j;
            }
        }
    }

    int ans1 = max_difference_2(arr);
    int ans2 = max_difference_3(arr);

    cout << max(max_val, max(ans1, ans2)) << endl;

    return 0;
}
```

Python Code:

```
def max_difference_2(arr):
    for i in range(len(arr)):
        arr[i] = arr[i] + i

    max_val = float('-inf')
    min_val = float('inf')

    for num in arr:
        if num > max_val:
            max_val = num

        if num < min_val:
            min_val = num

    for i in range(len(arr)):
        arr[i] = arr[i] - i
    return max_val - min_val

def max_difference_3(arr):
    for i in range(len(arr)):
        arr[i] = arr[i] - i

    max_val = float('-inf')
    min_val = float('inf')

    for num in arr:
        if num > max_val:
            max_val = num

        if num < min_val:
            min_val = num

    for i in range(len(arr)):
        arr[i] = arr[i] + i

    return max_val - min_val

def main():
    n = int(input())
    arr = list(map(int, input().split()))

    max_val = float('-inf')
    for i in range(n):
        for j in range(n):
            if abs(arr[i] - arr[j]) + i - j > max_val:
                max_val = abs(arr[i] - arr[j]) + i - j

    ans1 = max_difference_2(arr.copy())
    ans2 = max_difference_3(arr.copy())

    print(max(max_val, max(ans1, ans2)))

if __name__ == "__main__":
    main()
```