**CHALLENGE 4**

*Break an array loop*

**Objective**

We want to **understand** the algorithm below.

| INPUT | Array of 6 integers |
| --- | --- |
| OUTPUT | 0 (false) or 1 (true) |

// Input

int input[6] = {6, 10, 11, 45, 80, 82};

// Algorithm

bool isValid = true;

for (int i = 1; i < 6; i++) {

if (input [i] <= input [i - 1]) {

isValid = false;

break;

}

}

// Output

printf("%d", isValid);

**Q1 -** What do input[i] and input[i-1] represent?

| arr[i] | Represents the element in the array that is in arr[i] position |
| --- | --- |
| arr[i-1] | Represents the element is the array that is in arr[i-1]position |

**Q2 – Execute** mentally this code and write the output for each input

| INPUT | OUTPUT |
| --- | --- |
| {1, 2, 4, 7, 8, 9} | 1 |
| {1, 2, 7, 4, 8, 9}; | 0 |
| {1, 2, 3, 4, 5, 0}; | 0 |

**Q3 -** Understand the **goal** on this algorithm, by completing the comment below

The goal of this algorithm is to check if there’s number that’s bigger than the one after it

**Q4 -** Update the previous exercise code to avoid the usage of a BREAK inside the LOOP.

| EXAMPLE OF CODE WITH BREAK | EXAMPLE OF CODE WITHOUT BREAK |
| --- | --- |
| for (int i = 0; i < 10; i++) {  if (i == 5) {  break;  }  printf("%d ", i);  } | for (int i = 0; i < 5; i++) {  }  printf("%d ", i); |

// Input

int input[6] = {6, 10, 11, 45, 80, 82};

// Algorithm

bool isValid = true;

for (int i = 1; i < 6 && inValid; i++) {

if (input [i] <= input [i - 1]) {

isValid = false;

}

}

// Output

printf("%d", isValid);