Logo

STUDENT REPORT

DETAILS

Name

MOHAMMED FURKHAN

Roll Number

KUB23CSE085

EXPERIMENT

Title

ADVACED SUB ARRAY PROBLEM

Description

You are competing in a basketball contest. In this contest the score for each successful shot depends on both the distance from the basket and the player's position. The ball is shot N times, successfully. You are given an array A containing the distance of a player from basket for N shots. The index of array represents the position of the player. Score is calculated by multiplying the position with the distance from the basket.

Your task is to find and return an integer value, representing the maximum possible score you can achieve by choosing a contiguous subarray of size K from the given array.

Note:

* A subarray is a contiguous part of array.

* Assume 1 based indexing.

* The array contains both negative and positive values.

* Assume the player is standing on a cartesian plane.

Input Format

- **input1**:An integer value N representing the number of shots made by the player

- **input2** : An integer K representing the size of subarray

- i**nput3 :** An array of integers

Sample Input

```
5
2
1 2 3 4 5
```

Sample Output

```
14
```

Source Code:

```
def max_score(N, K, A):
    # Step 1: Calculate the scores based on the given distances
    scores = [(i + 1) * A[i] for i in range(N)]  # i+1 for 1-based index

    # Step 2: Find the maximum score for contiguous subarray of size K
    max_score = float('-inf')

    # Calculate the sum of the first K elements
    current_sum = sum(scores[:K])
    max_score = max(max_score, current_sum)

    # Use the sliding window technique
    for i in range(K, N):
        current_sum += scores[i] - scores[i - K]
        max_score = max(max_score, current_sum)

    return max_score

# Sample Input
N = 5
K = 2
A = [3, 4, 5]

# Function call
result = max_score(N, K, A)
print(result)  # Output: 14
```

RESULT

0 / 5 Test Cases Passed | 0 %