NAME: MUHAMMAD SHAMEER
SECTION: BCS-1A
ROLL NO: 25K-0744

Q1: Running Sum of 1D Array

```c
#include <stdlib.h>

int* runningSum(int* nums, int numsSize, int* returnSize) {
    *returnSize = numsSize;
    int* sumArr = (int*)malloc(numsSize * sizeof(int));
    if (!sumArr){
        return NULL;
    }
    sumArr[0] = nums[0];
    for (int i = 1; i < numsSize; i++) {
        sumArr[i] = sumArr[i - 1] + nums[i];
    }

    return sumArr;
}
```

Q2: Shuffle the Array

```c
int* shuffle(int* nums, int numsSize, int n, int* returnSize){
    *returnSize = numsSize;
    int *shuffledArr = (int*)malloc(numsSize * sizeof(int));
    if (!shuffledArr){
        return NULL;
    }

    for (int i = 0; i < n; i++){
        shuffledArr[2 * i] = nums[i];
        shuffledArr[2 * i + 1] = nums[n + i];
    }
    return shuffledArr;
}
```

NAME: MUHAMMAD SHAMEER
SECTION: BCS-1A
ROLL NO: 25K-0744

Q3: Transform Array by Parity

```c
#include <stdlib.h>

int* transformArray(int* nums, int numsSize, int* returnSize) {
    *returnSize = numsSize;
    int *transformedArr = (int*)malloc(numsSize * sizeof(int));
    if (!transformedArr){
        return NULL;
    }
    int zeroCount = 0;
    for (int i = 0; i < numsSize; i++){
        if (nums[i] % 2 == 0){
            zeroCount++;
        }
    }
    for (int i = 0; i < numsSize; i++){
        if (i < zeroCount){
            transformedArr[i] = 0;
        } else{
            transformedArr[i] = 1;
        }
    }

    return transformedArr;
}
```