

Instructions

- Work in this lab individually. Follow the best coding practices and include comments to explain the logic where necessary.
- You can use your books, notes, handouts, etc. but you are not allowed to borrow anything from your peer student.
- **Do not use any AI tool for help; doing so will be considered cheating and may result in lab cancellation and possible disciplinary action.**
- Test your program thoroughly with various inputs to ensure proper functionality and error handling.
- Show your work to the instructor before leaving the lab to get some or full credit.

Task 01

Implement a **recursive function** that **returns** the power x^n . Where x and n are parameters of the function.

```
double power (double x, int n);
```

Task 02

Implement a **recursive function** that **returns** the **sum of array elements**. The parameters **A** (the array) and **n** (its size) are passed to the function.

```
int arraySum (int A[], int n);
```

Example:

- Given the array $A = \{1, 2, 3, 4\}$ with $n = 4$, the function should **return** 10.

Task 03

Implement a **recursive function** that takes an **array** and the **starting** and **ending indices** of a portion of this array. The function should **reverse the contents of that portion** of the array. *You are not allowed to use any extra memory (e.g., another array):*

```
void reverseSubArray (int A[], int start, int end);
```

Example:

- Given $A = \{1, 2, 3, 4, 5, 6\}$ with $start = 1$ and $end = 4$, the array should become $\{1, 5, 4, 3, 2, 6\}$ after calling `reverseSubArray(A, start, end)`.

Task 04

Implement a **recursive function** that takes an array **A** containing **n** integers and an integer **k**. This function should **determine whether there exist two elements in A that sum to exactly k**. Your function should return **true** if such a pair exists and **false** otherwise.

```
bool checkSum (int A[], int start, int end, int k);
```

Parameters:

- **A[]**: The array of integers.
- **start** and **end**: The starting and ending indices of **A**.
- **k**: The target sum.

Initial Call: If **A** contains **n** integers, call `checkSum(A, 0, n-1, k)` with **k** as the desired **sum**.

Example:

- If the array contains $\{8, 5, 3, 7, 2\}$ and **k** is **11**, then your function should return **true** (since $8 + 3 = 11$).
- If **k** is **16**, the function should return **false**, as there are no two elements that **sum** to **16**.

Hints

1. While examining an element $A[i]$, what other number do you need to make the **sum** equal to **k**?
2. You will need to use the **recursive linear search** function we implemented in class.

Driver Function: Implement a driver function to test **checkSum** with the following behavior:

Enter the size of array: **5**

Enter the 5 elements of the array: **8 5 3 7 2**

Enter k: **11**

11 can be obtained by adding two elements of the array.

More (y/n)? **y**

Enter k: **16**

16 can NOT be obtained by adding two elements of the array.

More (y/n)? **n**