Data Structures and Algorithms Lab

Lab 09 **Marks 10**

Instructions

Work on this lab individually. You can use your books, notes, handouts etc. but you are not allowed to borrow anything from your peer student.

Marking Criteria

Show your work to the instructor before leaving the lab to get some or full credit.

What you must do

Program and **test** each of the following tasks in your C++ compiler.

Task 1

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 2

Implement a **recursive** function that **returns** the **sum** of array elements. Where **array** and its **size** are passed to the function as its parameters. The prototype of your function should be:

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

Task 3

Implement a **recursive** function which 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. The prototype of your function should be:

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

Task 4

Implement a **recursive** function which 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 there exist two elements in **A** whose sum is **k**, and it should return **false** otherwise. The prototype of your function should be:

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

Here, **start** and **end** are **starting** and **ending** indices of array **A**. If the array **A** contains **n** integers, then the initial function call will be `checkSum (A, 0, n-1, k)`, where **k** is the desired sum.

For example, if the array contains `{8, 5, 3, 7, 2}` and **k** is **11**, then your function should return **true**, because there exist two elements **(8 and 3)** which sum to **11**. However, if **k** is **16** then your function should have returned **false**, because there **do not exist** two elements in the above array which sum to **16**.

Hints:

- While looking at an element of the array **A[i]**, what other number do you need to make the sum equal to **k**?
- You will need the implementation of the **recursive linear search** function that we implemented in class.

Also implement a **driver function** to test your implementation. A sample run of your program should look like (text shown in dark red is entered by the user):

```
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
```

☺ ☺ ☺ **BEST OF LUCK** ☺ ☺ ☺