

Computer Science and Programming Lab Class 4

Task 1 *Function – Parameter passing* (10 minutes)

Implement the task of "find all prime numbers in the range [2,1000]" by using two functions: The first function judges if a given number is a prime number and the second function, given a range of numbers, return a list of all prime numbers in the range.

Task 2 *Function – Recursion* (5 minutes)

Please implement a function *recsum*(*N*) which computes the sum of all numbers from 1 to *N* recursively.

Task 3 *Function – Recursion* (15 minutes)

Please implement a function called *fibonacci*() which reads in an integer **n** and returns the **n**th number in Fibonacci sequence.

Hint: The Fibonacci sequence is characterized by the fact that every number after the first two is the sum of the two preceding ones. That is $F_n = F_{n-1} + F_{n-2}$. We define that $F(0) = 0$ and $F(1) = 1$. Please solve this problem using recursion, by breaking down a larger problem into smaller subproblems.

Task 4 *Recursion – Pascal's Triangle* (15 minutes)

Write a function which implements the Pascal's triangle. The first rows of the triangle are shown on the next page:

1																													
1					1																								
1			2			1																							
1		3		3		1																							
1		4		6		4		1																					
1		5		10		10		5		1																			
1		6		15		20		15		6		1																	
1		7		21		35		35		21		7		1															
1		8		28		56		70		56		28		8		1													
1		9		36		84		126		126		84		36		9		1											
1		10		45		120		210		252		210		120		45		10		1									
1		11		55		165		330		462		462		330		165		55		11		1							
1		12		66		220		495		792		924		792		495		220		66		12		1					
1		13		78		286		715		1287		1716		1716		1287		715		286		78		13		1			
1		14		91		364		1001		2002		3003		3432		3003		2002		1001		364		91		14		1	

Task 5 *Recursion – Full permutation*(15 minutes)

Write a function which receives a list and print all permutation of the elements, i.e., all possible orders of elements.

```
# sample input:
permutation([1,2,3])
# sample output:
[1, 3, 2]
[1, 2, 3]
[3, 1, 2]
[3, 2, 1]
[2, 1, 3]
[2, 3, 1]
```

Task 6 *Loop – Goldbach conjecture* (15 minutes)

Goldbach's conjecture is one of the oldest and best-known unsolved problems in number theory and all of mathematics. It states: **Every even integer greater than 2 can be expressed as the sum of two primes.**

Please verify Goldbach conjecture for integers smaller than 1000. For any even number n in that interval which obeys the conjecture, please print the two prime numbers summing up to n .

Hint: Find all prime numbers smaller than 1000 at first and store them in a list. Afterwards, iterate all numbers from 1 to 999, and try to express them as a sum of any two elements of the prime number list. You can generate all pairs of prime numbers by using a nested loop. However, note that a function without nested loops would be more efficient.

Task 7 *Function – Recursion* (20 minutes)

Write a recursive function `find_index()`, which returns the index of a number in the Fibonacci sequence, if the number is an element of this sequence and returns -1 if the number is not contained in it, i.e., we have: `fibonacci(find_index(n)) == n`