😋

Introduction to recursive programming

Recursion on natural numbers

1. Write a recursive function that computes the factorial of a positive number `N`.
2. Write a recursive function that computes the sum of the first `N` natural numbers.
3. Write a recursive function that counts the number of digits in a given natural number `N`.
4. Write a recursive function that counts the even digits in a given natural number `N`.
5. Write a recursive function that counts the odd digits in a given natural number `N`.
6. Write a recursive function that checks if a given natural number `N` contains only even digits.
7. Write a recursive function that checks if a given natural number `N` contains only odd digits.
8. Write a recursive function that returns the sum of the digits that comprise a given natural number `N`.
9. Write a recursive function that returns the product of the digits that comprise a given natural number `N`.
10. Write a recursive function that checks if a given natural number `N` contains duplicate digits.
11. Write a recursive function that converts a natural number `N` to its binary equivalent.
12. Write a recursive function that converts a natural `N` representing a binary number to its decimal equivalent.
13. Write a recursive function that checks if a given natural number `N` is Armstrong.

14. Write a recursive function that checks if a given natural number `N` contains a given digits d.
15. Write a recursive function that computes the power function.
16. Write a recursive function that returns the sum of the first `N`even numbers.
17. Write a recursive function that returns the sum of the first `N`odd numbers.
18. Write a recursive function that returns the reverse of a given natural number `N`.
19. Write a recursive function that returns the number of times a given digit appears in a given natural number `N`.
20. Write a program that calculates with how many zeroes the factorial of a given number ends.
21. Write a recursive function that determines if a given natural number is prime.
22. Write a recursive function that returns the maximum digit in a given natural number `N`.
23. Write a recursive function that returns the minimum digit in a given natural number `N`.
24. Write a recursive function that calculates the sum of the first `N`squares.
25. Write a recursive function that counts the number of digits greater than or equal to a given digit d in a given natural number `N`.
26. Write a recursive algorithm that computes the persistence of a given natural number `N`.
27. Write a recursive function that computes the logarithm of a given natural number `N` in a given base b.
28. Write a recursive function (Egyptian multiplication) that calculates the product of two positive integers `a` and `b` according to the Egyptian multiplication method.
29. Write two recursive functions div and mod which, given two positive integers `a` and `b`, return the quotient for one, and the remainder for the other of the division of `a` by `b` without using the division operator.
30. Write a recursive function that takes in a natural number `N` in decimal base and returns the number of bits set to 1 in its binary representation.
31. Write a recursive function that takes in two natural numbers `a` and `b` and returns their sum.

32. Write a recursive function that takes in two natural numbers `a` and `b` and returns their product.
33. Write two recursive functions, even and odd that determine if a given number is odd or even.

Double sums

1.

$$\sum_{i=1}^{m}\sum_{j=1}^{n}(i+j)$$

2.

$$\sum_{i=1}^{m}\sum_{j=1}^{n}(i^2 j^3)$$

3.

$$\sum_{i=1}^{m}\sum_{j=1}^{n} j$$

Recursion on Arrays

1. Describe a recursive algorithm that will check if an array A of integers contains an integer A[i] that is the sum of two integers that appear earlier in A, that is, such that A[i] = A[j] +A[k] for j,k < i.
2. Write a recursive function that will rearrange an array of `int` values so that all the even values appear before all the odd values.
3. Write a recursive function that finds the minimum and maximum values in an array of `int` values without using any loops.
4. Write a recursive function that determines if a string s is a palindrome, that is, it is equal to its reverse. For example, "racecar" and "gohangasalamiimalasagnahog" are palindromes.

5. Write a recursive procedure that displays the elements of an array `T` in reverse order to that of the array.
6. Write a recursive function for determining if a string `s` has more vowels than consonants.
7. Write a recursive function that calculates the sum of the positive elements of an array.
8. Write a recursive function that calculates the product of the elements of an array.
9. Write a recursive function that returns the maximum number of elements of an array.
10. Write a recursive function min_max that to a given array (of distinct integers) associates the pair of indices of the smallest element and the largest element. By For example, min_max [7, 2, 9, 3, 12, 4] returns: 2, 5.

Recursion on Matrices

1. Write a recursive function that returns the maximum number in a matrix of numbers.
2. Write a recursive function that returns the minimum number in a matrix of numbers.
3. Write a recursive function that returns the sum of numbers in a matrix of numbers.
4. Write a recursive procedure that returns the product of numbers in a matrix of numbers.
5. Write a recursive procedure that adds two given matrices.
6. Write a recursive procedure that computes the product of two matrices.
7. Write a recursive function that determines if all rows of given matrix are sorted in ascending order.
8. Write a recursive procedure that computes the transpose of a given matrix `M`
9. Write a recursive function that determines if all numbers stored in a given matrix `M` are even.

10. Write a recursive function that determines if all numbers stored in a given matrix `M` are odd.