4. Arrays

4.01. Write a function which takes two arguments - an array and its size and prints all elements of the array on the console.

4.02. Write a function which takes two arguments - an array and its size and reverses the array. Example: {4,2,8,-6,12}->{12,-6,8,2,4}.

4.03. Write a program which asks the user to enter n - the number of numbers he will enter. Then the program should ask for n numbers and print the contents of the array in reverse.

Example input	Expected output
4 2 4 6 8	8 6 4 2
5 1 0 1 0 -1	-1 0 1 0 1

4.04. Write a program which asks the user to enter n - the number of numbers he will enter. Then the program should ask for n numbers and print only the even numbers.

Example input	Expected output
4 1 2 3 -4	2 -4
5 1 3 0 7 9	0

4.05. Write a program which asks the user to enter n - the number of numbers he will enter. Then the program should ask for n numbers and print only the numbers that are smaller than the average of all numbers.

Example input	Expected output
7 1 2 3 4 5 6 7	1 2 3
5 -2 4 -3 6 8	-2 -3

4.06. Write a program which asks the user to enter n - the number of numbers he will enter. Then the program should ask for n numbers and print the maximum and minimum of them.

Example input	Expected output
6 -2 -1 0 1 3 4	4 -2
1 4	4 4

4.07. Write a function which takes three arguments - two double arrays - *source* and *destination* and unsigned *s* - size of the source. The function has to copy all elements from *source* to *destination* (assume destination is big enough).

4.08. Write a program which inputs a sequence of N integer numbers $a_1, a_2, \ldots, a_N$ and finds and prints the sum of those elements from the sequence, which can be representet as doubled odd number.

4.09. Write a program which cheks whether there exist two consequtive zeroes in a given sequence of integers.

4.10. Two sequences of numbers are given. Write a program that detects how many times the second sequence contains the first one.

4.11. Each subsequence of equal numbers in an array is called a train. Write a program that finds the beginning and length of the longest train in an array.

4.12. Write a program which check whether an array of chars is symmetric.

4.13. Write a function which takes two arguments - an int array and its size, and sorts the array (in ascending or descending order).

4.14. (*Cartesian Product*) Write a program which reads an array from the console and prints the product of each of its elements with all elements. E.g. for the array {1, 7, 3}, the result would be {1 * 1, 1 * 7, 1 * 3, 7 * 1, 7 * 7, 7 * 3, 3 * 1, 3 * 7, 3 * 3}, which gives us the array {1, 7, 3, 7, 49, 21, 3, 21, 9}, so for the input {1 7 3} the program should print 1 7 3 7 49 21 3 21 9.

Example input	Expected output
3 1 7 3	1 7 3 7 49 21 3 21 9
2 -1 4	1 -4 -4 16
1 42	1764

4.15. (*Closest Numbers*) Write a program which finds the two closest (by value) integer numbers in an array and prints the absolute difference between them.

Example input	Expected output	Explanation
5 1 105 10 100 3	2	The closest number are 1 and 3, |1-3|=2
8 1 2 3 4 5 6 7 8	1	All numbers are exactly 1 unit apart
4 -1 -10 -90 3	4	The closest numbers are -1 and 3, |-1-3|=4, or equivalently |3-(-1)|=|3+1|=4

4.16 Write a function which takes two arguments - an int array and its size, and returns its median.

4.17 Write a function which takes two arguments - an int array and its size, and returns its mode(s). "vector" and "algorithm" libraries are allowed, because the array can have more than one mode.

Example input	Expected output
1 2 3 4 4 5	4
22 3 11.9 22 3 4.5 -12 3	22 3
0 1 2 3	no mode

Mode & Median of a set of data

Type	Description
Median	Middle value separating the greater and lesser halves of a data set
Mode	Most frequent value in a data set

[More about Median and Mode of a set of data values here] (https://www.khanacademy.org/math/statistics-probability/

summarizing-quantitative-data/mean-median-basics/a/mean-median-and-mode-review "Markdown Tutorial")

4.18. Write a program which asks the user to enter n - the number of numbers he will enter. Then the program should ask for n numbers and determine if the given array is hacksaw. We call a sequence of numbers hacksaw if the following is true: $a_0 < a_1 > a_2 < a_3 > a_4 < a_5 \dots a_n$ or $a_0 > a_1 < a_2 > a_3 < a_4 > a_5 \dots a_n$

Example input	Expected output
3 **1 4 1**	yes
3 **4 1 4**	yes
6 **1 3 2 8 9 4**	no
7 **0 1 0 1 0 1 0**	yes
4 **4 -8 19 22**	yes
4 **4 -8 19 19**	no

4.19. Write a function with signature bool checkDate(const char* text), which verifies if the input parameter is a correct date in DD.MM format. Do not use string.

4.20. Write a function which takes four arguments - an int array, first index and last index of the searched interval and wanted value x. Using the algorithm binary search, the function should return the index of x if it exists in the array, or -1 if it does not exist.

4.21. Write a program which prints on the console the elements of the longest strictly increasing subsequence in a locally defined array. In case there are two sequences with the same length, print the first one.

Example input	Expected output
-4 -2 8 9 0 1 2 3 -4 2	-4 2 8 9
3 9 0 2 1 3 4 6 9	1 3 4 6 9

4.22. Write a function which prints all the combinations of elements from an array of integers arr, with sum equals to a natural number N.

4.23. Write a function with signature size_t *checkNums(long num1, long num2)* , which returns a pointer to an array containing all the digits encountered both in num1 and num2.

4.24. Write a bool function which checks whether a set is a subset of another set.

4.25. Write a program that finds the most frequent number in a given sequence of numbers. In case of multiple numbers with the same maximal frequency, print all of them, ordered from smallest to largest, separated by spaces.

Example input	Expected output	Explanation
12 **4** 1 1 **4** 2 3 **4** **4** 1 **4** 9 3	**4**	The number **4** is the most frequent (occurs 5 times)
8 **2 2 2 2** 1 **2 2 2**	**2**	The number **2** is the most frequent (occurs 7 times)
11 **7 7 7** 0 **2 2 2** 0 **9 9 9**	**2 7 9**	The numbers **2, 7** and **9** have the same maximal frequency (each occurs 3 times)

4.26. Given two sequences (of numbers) $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_m$ sorted in ascending order. Write a function that merges the two sequences in one new sequence with ascending ordered elements $c_1, c_2, \ldots$ consisting of the elements of the first two sequences. Do not use sorting algorithms. Use the condition that the first two sequences are sorted in ascending order. T*he complexity of your function must be* $\Theta(n+m)$.

4.27. In number theory, a Carmichael number is a composite number n which satisfies the modular arithmetic congruence relation: $b^n \equiv b(mod_n)$ for all integers b which are relatively prime to n. They are named after [Robert Carmichael](https://en.wikipedia.org/wiki/Robert_Daniel_Carmichael "Markdown Tutorial"). An alternative and equivalent definition of Carmichael numbers is given by [Korselt's criterion](https://en.wikipedia.org/wiki/Carmichael_number "Markdown Tutorial"): A positive composite integer n is a Carmichael number if and only if n is square-free (a square-free integer is an integer which is divisible by no perfect square other than 1), and for all prime divisors p of n, it is true that $p-1 \mid n-1$. Write a program that checks if a positive integer is a Carmichael number using Korselt's criterion.

4.28. Write a program which finds the sum of the polynomials $P_n(x) = a_n . x^n + a_{n-1} . x^{n-1} + \ldots + a_1 . x + a_0$ and $Q_m(x) = b_n . x^n + b_{n-1} . x^{n-1} + \ldots + b_1 . x + b_0$.

4.29. Write a program which finds the product of the polynomials $P_n(x) = a_n . x^n + a_{n-1} . x^{n-1} + \ldots + a_1 . x + a_0$ and $Q_m(x) = b_n . x^n + b_{n-1} . x^{n-1} + \ldots + b_1 . x + b_0$.

4.30. Write a program which finds the k-th power of the polynomial $P_n(x) = a_n . x^n + a_{n-1} . x^{n-1} + \ldots + a_1 . x + a_0$, where $k = 0,1,2,...$ is a given number.

github.com/andy489