1. Declare 4 variables, assign them arbitrary numerical values. Print the sum of the first and third variables, then print the result of dividing the second and fourth variables.

2. Enter 3 numbers. If they are all greater than 10, print their sum. Otherwise, check which of them is the largest and then multiply it by itself and then print their sum.

3. Ask the user to enter his year of birth. Calculate the user's age and print the corresponding message ('You are of legal age'/'You are not of legal age'). Do not take into account if the user enters an invalid year (e.g. 2025) - in that case print the corresponding message.

4. Ask the user to enter the year of production of his car. Calculate the age of the car and if it is older than 30 years, write 'Oldtimer', and if it is younger than 5 years, write 'Newbie'. Do not take into account if the user enters an invalid year (eg 2025) - in that case print the appropriate message.

5. Ask the user to enter his name, surname and number of years. If the user is under 20, write: First name + last name + ", you are very young!". If the user is under 30, write: First name + last name + ", you are young!". If the user is between 30 and 45, write: First name + last name + ", u

you are in the best years! ". Do not comment on elderly people. Do not take into account if the user enters an invalid year (eg 2025) - in that case print the appropriate message.

6. Ask the user to enter the name and number of years. Repeat this 3 times. If there are 2 people of the same age, write their names. Otherwise, check if there are 2 people with the same name, if there are, print their name and the sum of years. If neither is satisfied, print that there is no namesake or peer!

7. Ask the user to enter 3 ratings. If one of them is 1, write "You are not satisfied". Otherwise, add all 3 grades and calculate the average. If the average is less than 2.5, write "Sufficient 2", for an average between 2.5 and 3.5 write "Good 3", for an average between 3.5 and 4.5 write "Very

good 4", for an average above 4.5 write "Excellent 5".

8. Ask the user to enter the answer to the question "Are you afraid of the dentist?" If the answer is negative, write "You are brave!!", otherwise, write "Me too!!". For other answers, print that the entry is incorrect. Solve this task using switch-case and if-else.

9. Ask the user to enter a number smaller than 25. Using the switch-case, check whether the entered number is a multiple of 5 (these are the numbers 5, 10, 15 and 20). If yes, print the entered number, if not, print that the number is not a multiple of 5. For all other numbers that are not between 0 and 20, print that the wrong number was entered.

10. If the day of the week is entered, use the switch case to print: if it's Monday, Tuesday, Wednesday or Thursday, "When will it be the weekend?", for Friday, "It's Friday, Friday. Gotta get down on Friday", if it's Saturday or Sunday "Hey, it's the weekend!". If something else was entered, print "Wrong entry".

11. Ask the user to enter some text. Calculate how many words there are in the entered text.

12. Enter the whole number. After that, add up all the numbers from 1 to the entered number, and print the result.

13. Ask the user to enter a number less than 20. Then print all the numbers from the entered number up to 1. If a larger number has been entered, print the entered number.

14. Ask the user to enter a number. If it is even, print all the numbers backwards from the entered number to 1, otherwise, add all the numbers from 1 to the entered number.

17. Request to enter a two-digit number. Print with which numbers from 1 to 10 it is divisible. If it is not two-digit, you just need to print the corresponding message.

18. Ask the user to enter a string. Print how many vowels there are in that string. Print the first 5 letters of the entered string.

19. Ask the user to enter a string. Print how many consonants there are in that string. Print the first 3 letters of the entered string.

20. Load 2 integers, using the prompt function and calculate their sum, difference, product and quotient, print the sum of all even numbers between those 2 numbers, and the sum of all odd numbers between them, not including them.

21. Load 5 whole two-digit numbers. If the entered number is not two-digit, discard it. Add the last digits of each entered number and print the sum.

22. Ask the user to enter numbers until he enters a number that is greater than 20. When he enters that number, print from which attempt a number greater than 20 was entered.

23. Using a while loop, allow the user to enter exactly 8 numbers. Print the fourth entered number and the sum of the first and fifth entered numbers.

24. Load numbers until the sum of entered numbers reaches 100. At the end, print how many numbers the user has entered in total.

25. Create a program that allows the user to enter numbers until he enters the number 0. The program should print how many positive and how many negative numbers the user has entered.

26. Ask the user to enter 2 numbers. Using a do-while loop to print odd numbers that

are between those two numbers.

27. Use a loop to fill in a sequence of 10 members, and print the sequence.

28. Declare an empty list and ask the user to enter numbers until the sum of the entered members

list does not exceed 50. After that, print the length of the list.

29. If you have a sentence: "Send grandmothers to sipke" remove all the vowels from it and write the sentence

without them. You can save that in a new variable. Then remove all the consonants from the sentence

and print it too.

30. Given a list of 10 real numbers: var [87, 93, 11, 27, 38, 100, 42, 31, 8, 40] Find the number closest to zero and the number farthest from zero.

31. Ask the user to enter numbers in the list until he enters 0. Print the first middle and last element of the sequence.

32. Load numbers into the list until their sum exceeds 150. Create a new list composed of

odd numbers from the first list. Print a new list sorted from largest to smallest.

33. Load numbers into the list until their sum exceeds 250. Create a new list consisting of

even numbers from the first list. Print a new list sorted from smallest to largest.

34. Declare a list consisting of 5 words. Create a new list consisting only of those words

from the first list that have more than 2 letters. Print a new string of words.

35. List 5 words. Print the longest and shortest word from the list.

36. Load a list of 10 numbers. Print how many even and how many odd numbers are in the list. Then

print the average value of even numbers. Also print which sum is greater: sum of even numbers or sum of odd numbers.

37. Load a list of 10 numbers. Print the largest and smallest member of the list, then order the members of the list

by size - from the largest to the smallest and print it, then from the smallest to the largest and

also print it.

38. Ask the user to enter 2 numbers. If the first number is greater than the second, call the function

which will subtract the second from the first. Otherwise, call a function that will sum them. Print the result

in both cases.

39. Simple calculator - Ask the user to enter two numbers and a sign for the desired operation

do over the numbers (+, -, *, /). Write a function that imitates the operation of a simple calculator: receives

two numbers, and an operation sign. Depending on the sign, perform the requested operation. The function returns

the result to the main program, then it prints it.

40. Ask the user a simple mathematical expression (eg: How much is 2+2?). In function to check

accuracy of the entered answer. If true, print a message and return true. If not correct, return

correct result to the main program, where the message "Your answer is incorrect,

the correct answer is: ...".

41. Create a function that returns the number of days in that month to the forwarded month. Print the result in the main program.

42. Use the function to check whether the year entered by the user is a leap year or not.

43. Write a function that will require the user to enter 5 numbers and save them in an array (make sure there are exactly 5). The "check()" function should check all elements from the array and only multiply even negative ones by themselves. Print a string in the main program.

44. Write a function that will calculate the tax on the amount paid. Tax rate for amounts between

100 and 500 is 1%, between 500 and 1000 is 5%, and over 1000 is 10%. The function receives as a parameter

amount, and returns the tax amount.

45. Write a function that will return the circulation for the number of plates sold. If the plate is sold in multiples

out of 100,000 copies, 'silver', over 200,000 'gold' and over 500,000 'platinum' should be returned.

The function receives the number of copies sold.

46. Write a function that will return the number of colons and spaces for the passed text.

47. Write a function that will print the passed sentence backwards, but also replace uppercase and lowercase

letters. Call the function in the main program.

48. Ask the user to enter some text. Pass the text to a function that will check if the entered text is a palindrome. (A palindrome is a word that reads the same backwards and forwards.)

49. Write a function that will receive the height in centimeters as a parameter and that will return a string

which contains height expressed in meters and centimeters. Call that function and print the result.

(If the parameter is 178, the function should return "1m and 78cm") .

50. Write the function "convertToSeconds()" which will convert the passed hours into seconds. The con version is done gradually, in such a way that within this function, the function "convertToM inutes()" will be created, which will first convert the sent hours into minutes and then return it to the main function, which will finish the conversion.

51. Print all properties of the object and their values:

public class Person {

public string Name {get;set;} public float Rating {get;set;} public string Author {get;set;}

}

Person a = new Person{ Name = 'The Wire', Rating = 9.4,

Author = 'David Simon'

};

52. Declare the object "building" and property-e address, building number, number of floors, number of apartments. Ask the user to enter property values and print the assigned values: "I live in building number" +building number+"at the address"+address+"which has"+number of floors+"floors and total" '+number of apartments+"apartments."

53. Declare the "test" object that has the properties: achieved_points, max_points and grade. Add a method that calculates the grade based on the amount of achieved and maximum points for that test.

54. Declare the object "person" which will have the properties name, surname and kilograms. Print the values you entered. Then declare a method that will calculate how many kilograms that person has on Mars. Pass the number of kilograms you entered to the method. Print the values you set again. (The acceleration of gravity on Mars is 0.38 of the acceleration of gravity on Earth.)

55. Ask the user to enter properties for the "city" object. It is necessary to enter the name of the city, the number of inhabitants, the area and the mayor. For the name of the city and the mayor, check that no numbers have been entered, and for the number of inhabitants and area, check that only numbers have been entered. Before assigning a value to the area, add 'km2' and then print everything related to the "city" object.

56. Create a "time" object that will have the properties hours, minutes and seconds to which you will assign arbitrary values. Also create methods: "convertHours", which will return the assigned hours converted to minutes, "convertMinutes" which will return the assigned minutes (including the hours we converted to minutes), converted to seconds, "convertSeconds" which will return the total number seconds and the "print()" method, which will print the time in the form of HH:MM:SS.

57. Declare an object that will contain a person's name, surname, profession, year_start_of_work, year_of_service and salary. The object should have an "increase" method that will add 50KM to the salary every time the seniority is increased by a year (calculate track_year as the difference from the current year to the start_year; the "increase" method should be called in the "track" method every time it is calculated that the seniority has increased by a year).

58. Declare the "person" object. Ask the user to enter values for the following 3 properties of the object: jmbg, first and last name. Write a function that checks for first and last name - they must not be shorter than 3 characters and must not contain numbers, and a function for JMBG - must not contain anything but numbers and must have exactly 13 numbers. If the check did not pass, the function should print an error message and return 0, if the check passed, the function should return 1, and only then should the entered values be assigned to the properties of the object. After each entry, call the appropriate function.

59. Declare a color object that will have the properties R, G, B initially set to 0, and the method

"set_color" which will set a randomly selected shade of color (0.255) to the defined properties. Call the method and print the resulting color value in the form (R,G,B).

60. Declare a color object that will have the properties R, G, B initially set to 0, and the method

"set_color" which will set a randomly selected shade of color (0.255) to the defined properties. Call the method and print the resulting color value in the form (R,G,B).

61. Declare the object "user" with the properties first name, last name, number_year, date_of_birth and fill it with arbitrary values. Write the function "birthday()" which, according to the passed object, checks whether today is the user's birthday, if it is, it prints "Happy birthday!", if not, it returns 0.

62. Write the "print()" function, which will receive the weight in grams as a parameter and return a string containing the weight expressed in kilograms and grams. (If the parameter is 1981, the function should return "1kg and 981g"). Declare an object that will contain a person's height, weight, occupation and name.

The object should have an "add_mass" method that will change the person's weight, but so that it receives the weight in grams. The method calls the "print()" function and thus prints the new weight of the person in kilograms and grams. Call the "add_mass" method.

63. Write the "grade()" function that returns the appropriate grade for the entered percentage of correct results. Between 40 and 60% grade is 2, 61 and 75% grade 3, 76 and 90 grade 4, over 90% grade is 5. Declare an object that will contain the name of the student, the list of obtained points of completed tasks [1, 0, 2, 3 , 0, 0] (maximum number of points is 10), percentage and grade still unknown. The object should have a "points" method that will calculate and set the percentage of points on the test. The "points" method calls the function to calculate the grade. Using the "grade()" function, calculate the grade on the test and, using it, set the value of the "grade" property.

64. A JSON object is given:

17. string jsonObj =
{ “phones”:[
{ “id”:6144,
“name”:”Sony Xperia Z2”,
“manufacturer”:”Sony”,
“ram”:”3 GB”,
“internal_memory”:”16 GB”,
“cpu”:”Quad-core 2.3 GHz Krait 400”,
“back_camera”:”20.7 MP”,
“front_camera”:”2.2 MP”,
“gpu”:”Adreno 330”,
“battery”:”3200 mAh”,
“battery_removable”:”no”,
“price”:”490 EUR”
},
{ “id”:6033,
“name”:”Samsung Galaxy S5”,
“manufacturer”:”Samsung”,
“ram”:”2 GB”,
“internal_memory”:”32 GB”,
“cpu”:”Quad-core 2.5 GHz Krait 400”,
“back_camera”:”16 MP”,
“front_camera”:”2 MP”,
“gpu”:”Adreno 330”,
“battery”:”2800 mAh”,
“battery_removable”:”yes”,

“price”:”430 EUR”
},
{“id”:5705,
“name”:”LG Nexus 5”,
“manufacturer”:”LG”,
“ram”:”3 GB”,
“internal_memory”:”16 GB”,
“cpu”:”Quad-core 2.3 GHz Krait 400”,

“back_camera”:”8 MP”,
“front_camera”:”1.3 MP”,
“gpu”:”Adreno 330”,
“battery”:”2300 mAh”,
“battery_removable”:”no”,
“price”:”260 EUR”
},
{“id”:6077,
“name”:”LG G2 mini”,
“manufacturer”:”LG”,
“ram”:”1 GB”,
“internal_memory”:”8 GB”,
“cpu”:”Quad-core 1.2 GHz Cortex-A7”,
“back_camera”:”8 MP”,
“front_camera”:”1.3 MP”,
“gpu”:”Adreno 305”,
“battery”:”2440 mAh”,
“battery_removable”:”yes”,
“price”:”230 EUR”
},
{“id”:5497,
“name”:”Sony Xperia M”,
“manufacturer”:”Sony”,
“ram”:”1 GB”,
“internal_memory”:”4 GB”,
“cpu”:”Dual-core 1 GHz Krait”,
“back_camera”:”5 MP”,
“front_camera”:”VGA”,
“gpu”:”Adreno 305”,
“battery”:”1750 mAh”,
“battery_removable”:”yes”,
“price”:”110 EUR”
},
{“id”:3724,
“name”:”Samsung Galaxy Ace”,
“manufacturer”:”Samsung”,
“ram”:”512 MB”,
“internal_memory”:”158 MB”,
“cpu”:”800 MHz ARM 11”,
“back_camera”:”5 MP”,
“front_camera”:”no”,
“gpu”:”Adreno 200”,
“battery”:”1350 mAh”,
“battery_removable”:”yes”,
“price”:”90 EUR”
}
]
};

Write the following functions:

a) Write the name of the most expensive mobile phone.

b) Print out all mobile phones that have a removable battery.

c) Write the function "changePrice()" which enables entering a new price (from the user) and changing the price to the forwarded mobile phone. At the same time, make sure that the user enters an acceptable price (it must consist only of numbers and the new price must be lower or higher only by 0% to 20% of the old price). As long as the new price does not pass the check, print the corresponding message and ask the user to re-enter it. When saving a new price to the facility, you should add the currency "EUR" to the end of the price.

d) Write the function "avgRam()" which calculates the average amount of mobile phone RAM and prints it in GB and MB.

e) Write the "sortById()" function, which should sort the mobile phones by ID from the smallest to the largest, then print them.

f) Create a class "Smartphone" which consists of properties: name, manufacturer, internal_memory, price and discount. Instantiate Smartphone class objects for each cell phone from the default object "jsonObj" (do this dynamically, not just copy-paste property values from the default object).

g) Add the "setDiscount" method, which will increase the discount by the received parameter

10. The method calls the "printPrice()" function, which prints the price including the discount.

65. ATM simulation