

Elementary

1. Write a program that prints 'Hello World' to the screen.
2. Write a program that asks the user for their name and greets them with their name.
3. Modify the previous program such that only the users Alice and Bob are greeted with their names.
4. Write a program that asks the user for a number n and prints the sum of the numbers 1 to n .
5. Modify the previous program such that only multiples of three or five are considered in the sum, e.g. 3, 5, 6, 9, 10, 12, 15 for $n=17$.
6. Write a program that asks the user for a number n and gives them the possibility to choose between computing the sum and computing the product of $1, \dots, n$.
7. Write a program that prints a multiplication table for numbers up to 12.
8. Write a program that prints *all* prime numbers. (Note: if your programming language does not support arbitrary size numbers, printing all primes up to the largest number you can easily represent is fine too.)
9. Write a guessing game where the user has to guess a secret number. After every guess the program tells the user whether their number was too large or too small. At the end the number of tries needed should be printed. It counts only as one try if they input the same number multiple times consecutively.
10. Write a program that prints the next 20 leap years.
11. Write a function that returns the largest element in a list.
12. Write function that reverses a list, preferably in place.
13. Write a function that checks whether an element occurs in a list.
14. Write a function that returns the elements on odd positions in a list.
15. Write a function that computes the running total of a list.
16. Write a function that tests whether a string is a palindrome.
17. Write three functions that compute the sum of the numbers in a list: using a `for`-loop, a `while`-loop and recursion. (Subject to availability of these constructs in your language of choice.)
18. Write a function `on_all` that applies a function to every element of a list. Use it to print the first twenty perfect squares. The perfect squares can be found by multiplying each natural number with itself. The first few perfect squares are $1*1=1$, $2*2=4$, $3*3=9$, $4*4=16$. Twelve for example is not a perfect square because there is no natural number m so that $m*m=12$. (This question is tricky if your programming language makes it difficult to pass functions as arguments.)

19. Write a function that computes the list of the first 100 Fibonacci numbers. The first two Fibonacci numbers are 1 and 1. The $n+1$ -st Fibonacci number can be computed by adding the n -th and the $n-1$ -th Fibonacci number. The first few are therefore 1, 1, $1+1=2$, $1+2=3$, $2+3=5$, $3+5=8$.
20. Write a function that takes a number and returns a list of its digits. So for 2342 it should return [2,3,4,2].
21. Write a function that takes a list of strings and prints them, one per line, in a rectangular frame. For example the list ["Hello", "World", "in", "a", "frame"] gets printed as:

```

*****
* Hello *
* World *
* in    *
* a     *
* frame *
*****

```

22. Given two strings, write a program that efficiently finds the longest common subsequence.
23. Write a program in C to read 10 numbers from the keyboard and find their sum and average
24. Write a program in C to display the cube of the number up to an integer
25. Write a program in C to display the multiplier table vertically from 1 to n
26. Write a C program to check whether a given number is an Armstrong number or not
27. Write a C program to determine whether a given number is prime or not
28. Write a C program to display Pascal's triangle

```

      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1

```

29. Write a C program to check whether a number is a Strong Number or not
30. Write a program in C to get the largest element of an array using the function.