# The game of banknotes

# The game

This is a simple game, which aims to identify the smallest number of banknotes that allow reaching a certain amount in euros.

The amounts vary between 100 € and 1000 €, always multiples of 5, in order to be able to pay this amount only with notes. The notes in the game are:

* 50 € notes
* 20 € notes
* 10 € notes
* 5 € notes

Thus, the player should think about a solution that starts with the highest number of higher value notes, followed by decreasing value notes, until the remaining value is completed. Note the following example:

795 € = 50 € x 15 + 20 € x 2 + 5 € x 1

Which gives a total of 18 banknotes.

How to calculate the maximum number of notes for each amount? With the integer division...



With 15 banknotes of 50 € we are able to achieve 750 €, only missing 45 € for the total amount . The same process is repeated with 20 € banknotes...



With 2 banknotes of 20 €, we only miss 5€, wich can be addressed by just one banknote of 5 €.

Final result: 15 notes of 50 € + 2 notes of 20 € + 1 note of 5 €.

|  |  |
| --- | --- |
| **Balloon animal** | *Try this reasoning for these other amounts:*  *100 €, 240 €, 280 €, 335 €* |

So let's start by outlining our algorithm...

The algorithm will be as follows:

1. Define a random amount to be played.

2. Read the player's play: s/he must indicate the smallest number of notes that allow reaching the randomly set amount.

3. Calculate the number of notes:

3.1. banknotes ← subproblem calculate 50 € banknotes

3.2. banknotes ← banknotes + subproblem calculate € 20 banknotes

3.3. banknotes ← banknotes + subproblem calculate € 10 banknotes

3.4. banknotes ← banknotes + subproblem calculate € 5 banknotes

4. Check if the player got it right and present the result.

Note that we have several subproblems for which we will have to design the algorithm in sufficient detail. As we saw earlier, the calculation of the number of notes is always similar - integer division - varying only the divisor. We will then develop a function that solves this problem:

def notes(value, note):

n = value // note

if n > 0:

print(n, "notes of", note)

return n

* The function has two parameters, the total amount to be paid and the note value;
* determines the quotient of the integer division by the note value. If greater than zero, it displays the number of notes;
* at the end the function returns the number of notes.

Let's then create the function game() that implements this algorithm:

import random

def game():

# 1. amount to guess

amount = 5\*random.randint(20, 200)

print("How many banknotes are needed to pay for", amount, "€?")

# 2. Read the players' play

notes\_player = int(input())

# 3. Calculate the number of banknotes

notes\_50 = notes(amount, 50)

amount -= 50 \* notes\_50

notes\_20 = notes(amount, 20)

amount -= 20 \* notes\_20

notes\_10 = notes(amount, 10)

amount -= 10 \* notes\_10

notes\_5 = notes(amount, 5)

amount -= 5 \* notes\_5

# 4. Check if the player has guessed

if notes\_player == notes\_50 + notes\_20 + notes\_10 + notes\_5:

print("You won!")

else:

print("No. You needed", notes\_50 + notes\_20 + notes\_10 + notes\_5, "notes")

Function [random.randint](https://docs.python.org/3/library/random.html#functions-for-sequences)() select a random integer between a minimum and a maximum number.

|  |  |
| --- | --- |
| Head with gears | *Note the arguments in the call to* random.randint()*: 20 e 200. Justify the use of these values and not others.* |

In this case, multiples of 5 € between 100 € and 1000 € were chosen, so the limits of the set of possible integers range from 20 to 200.

The reading of the play by the human player is performed through the function [input()](https://www.w3schools.com/python/ref_func_input.asp).

In step 3 of the algorithm, four calls are made to the function notes(), with the 4 types of notes available.

And finally, in step 4 of the algorithm, the human player's play is compared and the calculated value.

|  |  |
| --- | --- |
| **Balloon animal** | *Play this game.* |

|  |  |
| --- | --- |
| Head with gears | *Note that when we call the same function* notes() *4 times, we are reusing that code.*  *What are the advantages of this reuse?* |

# Final challenge

Now try to adapt your game to the calculation of any amount from 100 to 1000. In this case, in addition to the notes, you should also use coins...

|  |  |
| --- | --- |
| **Share with person** | Share **your** game! |