Homework 2

Amin Parchami, Kamran Akbar, Alireza Kavian

Alice is visiting wonderland after a long time. She has just realized that all the free games have become priced. She goes to the Coin Machine to get all the coins she needs for the games! The Coin Machine has only a few certain types of coins. Part of this machine is to use your assembly code!

As input, your code will get the value Alice is looking for and the value of the different types of coins. Then it has to determine how many of each coin is needed.

Your code must use a **greedy algorithm** to find out the answer. It uses the highest valued coin, as much as possible, before trying the smaller coins.

Input:

The first line of input contains a positive integer as the desired value. In the next line, there are several positive integers determining the value of each coin in the descending order, followed by a 0 indicating the end of input.

output:

In the first line, your code must print how many of each coin has been used in the same order the coins are given. In the second line, it must print the remaining value, which obviously will be 0 if the value is achieved and will be bigger than 0 otherwise.

Your code **must** comply with the following rules:

- You can only use the EAX, EBX, ECX, EDX, ESI and EDI registers.
- You are not allowed to use the memory/Data segment.
- You must use the read_int and print_int functions (from the textbook) for I/O.

- You can only use the commands you have learned so far in the class. You cannot use MUL, IMUL, DIV, IDIV, etc.
- You must not print extra outputs.

Please upload only the ".asm" file on **courses.kntu.ac.ir**.

Your code will be checked for similarity. In the case of cheating, the student will receive a **negative** point. It is **your responsibility** to protect your code. Your code **must** be able to run using the following **driver.c** file.

```
void asm_main();
int main() {

    asm_main();

    return 0;
}
```

Example:
Input:
960
1000 500 150 30 5 0

Output:
0 1 3 0 2
0

You can see (**0** * 1000) + (**1** * 500) + (**3** * 150) + (**0** * 30) + (**2** * 5) = 960
So the desired value has been reached and the second line should be 0.
But if the first line of input was 538(no change in the second line),
then the output would be:
0 1 0 1 1
3