Exercise 01

Write a program that prints the initials of each group member. Each name must be separated by a newline character (\n). Use only numbers to represent each letter in the code (e.g., instead of 'A', use the number 65).
Allowed functions: write

Exercise 02

Write a program that prints the ASCII numbers representing the initials of each group member. Separate each number with a space. Each group of numbers representing a name must be separated by a newline character (\n). Use only numeric representation of characters.
Allowed functions: printf

Exercise 03

Write a program that prints the data type table for the following C types:

- char
- unsigned char
- short int
- int
- unsigned int
- short long
- long
- unsigned long
- float
- double

The output must contain the type name, minimum value, and maximum value.

Print as you wish!

Just an example of output:

```
type    minimum maximum
char    -128    127
int     ...
```

Allowed functions: printf

BONUS: *Demonstrate overflow and underflow values*.

An output example:

```
type    minimum maximum underflow       overflow
char    -128    127     -128-1=127      127+1=-128
int     ...
```

Why this results occurs?

Exercise 04

Write a program that counts from 0 to 42. Each number must be separated by a newline character (\n).

Allowed functions: write

Exercise 05

Write a program that receives a number as an argument and prints a sequence approaching 42.

Examples:

```
Meu terminal$ ./42counter 40
40, 41, 42
```

```
Meu terminal$ ./42counter 45
45, 44, 43, 42
```

Allowed functions: write, atoi

Exercise 06

Write a program that receives a number as an argument and prints it with its digits reversed.

Examples:

```
Meu terminal$ ./invert 1234
4321
```

```
Meu terminal$ ./invert -1234
-4321
```

Allowed functions: printf, atoi

Exercise 07

Write a program that receives a sequence of characters and prints the input.

Allowed functions: write

Exercise 08

Write a program that receives a sequence of characters and prints the ASCII number for each character, separated by spaces.

Example:
`Meu terminal$ ./ascii_printer AB`
`65 66`
`Meu terminal$ ./ascii_printer abeSTADO`
`97 98 101 83 84 65 68 79`
`Meu terminal$ ./ascii_printer "abe STADO"`
`97 98 101 32 83 84 65 68 79`

Allowed functions: write, atoi