

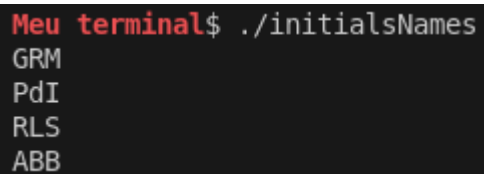
Exercise 01

Write a program that print the initials of each group member. For each member print all initials in one line (name and surname). Separate the names of the members with a newline.

In your code, all letters must be represented with the ASCII numbers of the letters (e.g. use 65 instead of 'A' and 10 instead of newline).

Example: The members:

- Gioconda Ronda Mujiro
- Pascoalito de Ilhita
- Renan Leite Silva
- Anabele Belle Bella



```
Meu terminal$ ./initialsNames
GRM
PdI
RLS
ABB
```

You can avoid printing the 'd', like in Pascoalito de Ilhita, if you want to. Or, print 'D'.

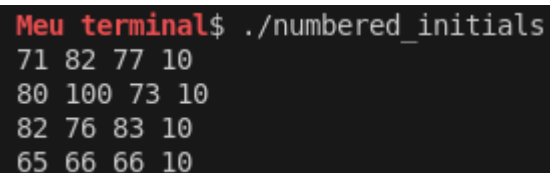
Allowed functions: write

Exercise 02

Write a program that prints the ASCII numbers representing the initials of each group member. The rules are analogous as before, the full initials name and separate them with a newline. Use only numeric representation of characters in the code.

Example:

The same names that before



```
Meu terminal$ ./numbered_initials
71 82 77 10
80 100 73 10
82 76 83 10
65 66 66 10
```

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
- 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.

Examples:

```
Meu terminal$ ./print_arg 'Uma frase'
Uma frase
Meu terminal$ ./print_arg Teste
Teste
Meu terminal$ ./print_arg
Meu terminal$ ./print_arg 'Mais uma frase'
Mais uma frase
Meu terminal$ ./print_arg 'Mais uma frase' kkkk
Meu terminal$
```

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 abeSTAD0
97 98 101 83 84 65 68 79

Meu terminal$ ./ascii_printer "abe STAD0"
97 98 101 32 83 84 65 68 79
```

Allowed functions: write, atoi