0x02. C - Functions, nested loops

**C**

* By: Julien Barbier
* Weight: 1
* Project over - took place from Sep 12, 2022 6:00 AM to Sep 15, 2022 6:00 AM
* An auto review will be launched at the deadline

In a nutshell…

* **Auto QA review:** 68.0/106 mandatory & 0.0/36 optional
* **Altogether:**  **64.15%**
  + Mandatory: 64.15%
  + Optional: 0.0%
  + Calculation:  64.15% + (64.15% \* 0.0%)  == **64.15%**

Resources

**Read or watch**:

* [Nested while loops](https://intranet.alxswe.com/rltoken/_4aLZ5nW24njUT2VbSZdQQ)
* [C - Functions](https://intranet.alxswe.com/rltoken/Vg1zzzrxLhPh71405uggSg)
* [Learning to Program in C (Part 06)](https://intranet.alxswe.com/rltoken/jveXtnJII2S0z7a06c7-JA) (*stop at 14:00*)
* [What is the purpose of a function prototype?](https://intranet.alxswe.com/rltoken/XZ--UJZO76ZoUWNA9bTmbg)
* [C - Header Files](https://intranet.alxswe.com/rltoken/AS8JW4ObD5gmyX2mgtqV0A) (*stop before the “Once-Only Headers” paragraph*)

Learning Objectives

At the end of this project, you are expected to be able to [explain to anyone](https://intranet.alxswe.com/rltoken/-ERX2Jh115GIdTUsSixnnA), **without the help of Google**:

General

* What are nested loops and how to use them
* What is a function and how do you use functions
* What is the difference between a declaration and a definition of a function
* What is a prototype
* Scope of variables
* What are the gcc flags -Wall -Werror -pedantic -Wextra -std=gnu89
* What are header files and how to to use them with #include

Requirements

General

* Allowed editors: vi, vim, emacs
* All your files will be compiled on Ubuntu 20.04 LTS using gcc, using the options -Wall -Werror -Wextra -pedantic -std=gnu89
* All your files should end with a new line
* A README.md file, at the root of the folder of the project is mandatory
* Your code should use the Betty style. It will be checked using [betty-style.pl](https://github.com/holbertonschool/Betty/blob/master/betty-style.pl) and [betty-doc.pl](https://github.com/holbertonschool/Betty/blob/master/betty-doc.pl)
* You are not allowed to use global variables
* No more than 5 functions per file
* You are not allowed to use the standard library. Any use of functions like printf, puts, etc… is forbidden
* You are allowed to use [\_putchar](https://github.com/holbertonschool/_putchar.c/blob/master/_putchar.c)
* You don’t have to push \_putchar.c, we will use our file. If you do it won’t be taken into account
* In the following examples, the main.c files are shown as examples. You can use them to test your functions, but you don’t have to push them to your repo (if you do we won’t take them into account). We will use our own main.c files at compilation. Our main.c files might be different from the one shown in the examples
* The prototypes of all your functions and the prototype of the function \_putchar should be included in your header file called main.h
* Don’t forget to push your header file

Copyright - Plagiarism

* You are tasked to come up with solutions for the tasks below yourself to meet with the above learning objectives.
* You will not be able to meet the objectives of this or any following project by copying and pasting someone else’s work.
* You are not allowed to publish any content of this project.
* Any form of plagiarism is strictly forbidden and will result in removal from the program.

More Info

You do not have to understand the call by reference (address), stack, static variables, recursions or arrays, yet.

Quiz questions

**Great!** You've completed the quiz successfully! Keep going! (Show quiz)

Tasks

0. \_putchar

**mandatory**

Score: 85.71% (*Checks completed: 85.71%*)

Write a program that prints \_putchar, followed by a new line.

* The program should return 0

julien@ubuntu:~/0x02$ gcc -Wall -pedantic -Werror -Wextra -std=gnu89 \_putchar.c 0-putchar.c -o 0-putchar

julien@ubuntu:~/0x02$ ./0-putchar

\_putchar

julien@ubuntu:~/0x02$

**Repo:**

* GitHub repository: alx-low\_level\_programming
* Directory: 0x02-functions\_nested\_loops
* File: 0-putchar.c

 Done? Help Check your code Ask for a new correction Get a sandbox QA Review

1. I sometimes suffer from insomnia. And when I can't fall asleep, I play what I call the alphabet game

**mandatory**

Score: 71.43% (*Checks completed: 71.43%*)

Write a function that prints the alphabet, in lowercase, followed by a new line.

* Prototype: void print\_alphabet(void);
* You can only use \_putchar twice in your code

julien@ubuntu:~/0x02$ cat 1-main.c

#include "main.h"

/\*\*

\* main - check the code

\*

\* Return: Always 0.

\*/

int main(void)

{

print\_alphabet();

return (0);

}

julien@ubuntu:~/0x02$ gcc -Wall -pedantic -Werror -Wextra -std=gnu89 \_putchar.c 1-main.c 1-alphabet.c -o 1-alphabet

julien@ubuntu:~/0x02$ ./1-alphabet

abcdefghijklmnopqrstuvwxyz

julien@ubuntu:~/0x02$

**Repo:**

* GitHub repository: alx-low\_level\_programming
* Directory: 0x02-functions\_nested\_loops
* File: 1-alphabet.c

 Done? Help Check your code Ask for a new correction Get a sandbox QA Review

2. 10 x alphabet

**mandatory**

Score: 71.43% (*Checks completed: 71.43%*)

Write a function that prints 10 times the alphabet, in lowercase, followed by a new line.

* Prototype: void print\_alphabet\_x10(void);
* You can only use \_putchar twice in your code

julien@ubuntu:~/0x02$ cat 2-main.c

#include "main.h"

/\*\*

\* main - check the code.

\*

\* Return: Always 0.

\*/

int main(void)

{

print\_alphabet\_x10();

return (0);

}

julien@ubuntu:~/0x02$ gcc -Wall -pedantic -Werror -Wextra -std=gnu89 \_putchar.c 2-main.c 2-print\_alphabet\_x10.c -o 2-alphabet\_x10

julien@ubuntu:~/0x02$ ./2-alphabet\_x10

abcdefghijklmnopqrstuvwxyz

abcdefghijklmnopqrstuvwxyz

abcdefghijklmnopqrstuvwxyz

abcdefghijklmnopqrstuvwxyz

abcdefghijklmnopqrstuvwxyz

abcdefghijklmnopqrstuvwxyz

abcdefghijklmnopqrstuvwxyz

abcdefghijklmnopqrstuvwxyz

abcdefghijklmnopqrstuvwxyz

abcdefghijklmnopqrstuvwxyz

julien@ubuntu:~/0x02$

**Repo:**

* GitHub repository: alx-low\_level\_programming
* Directory: 0x02-functions\_nested\_loops
* File: 2-print\_alphabet\_x10.c

 Done? Help Check your code Ask for a new correction Get a sandbox QA Review

3. islower

**mandatory**

Score: 85.71% (*Checks completed: 85.71%*)

Write a function that checks for lowercase character.

* Prototype: int \_islower(int c);
* Returns 1 if c is lowercase
* Returns 0 otherwise

FYI: The standard library provides a similar function: islower. Run man islower to learn more.

julien@ubuntu:~/0x02$ cat 3-main.c

#include "main.h"

/\*\*

\* main - check the code.

\*

\* Return: Always 0.

\*/

int main(void)

{

int r;

r = \_islower('H');

\_putchar(r + '0');

r = \_islower('o');

\_putchar(r + '0');

r = \_islower(108);

\_putchar(r + '0');

\_putchar('\n');

return (0);

}

julien@ubuntu:~/0x02$ gcc -Wall -pedantic -Werror -Wextra -std=gnu89 \_putchar.c 3-main.c 3-islower.c -o 3-islower

julien@ubuntu:~/0x02$ ./3-islower

011

julien@ubuntu:~/0x02$

**Repo:**

* GitHub repository: alx-low\_level\_programming
* Directory: 0x02-functions\_nested\_loops
* File: 3-islower.c

 Done? Help Check your code Ask for a new correction Get a sandbox QA Review

4. isalpha

**mandatory**

Score: 85.71% (*Checks completed: 85.71%*)

Write a function that checks for alphabetic character.

* Prototype: int \_isalpha(int c);
* Returns 1 if c is a letter, lowercase or uppercase
* Returns 0 otherwise

FYI: The standard library provides a similar function: isalpha. Run man isalpha to learn more.

julien@ubuntu:~/0x02$ cat 4-main.c

#include "main.h"

/\*\*

\* main - check the code.

\*

\* Return: Always 0.

\*/

int main(void)

{

int r;

r = \_isalpha('H');

\_putchar(r + '0');

r = \_isalpha('o');

\_putchar(r + '0');

r = \_isalpha(108);

\_putchar(r + '0');

r = \_isalpha(';');

\_putchar(r + '0');

\_putchar('\n');

return (0);

}

julien@ubuntu:~/0x02$ gcc -Wall -pedantic -Werror -Wextra -std=gnu89 \_putchar.c 4-main.c 4-isalpha.c -o 4-isalpha

julien@ubuntu:~/0x02$ ./4-isalpha

1110

julien@ubuntu:~/0x02$

**Repo:**

* GitHub repository: alx-low\_level\_programming
* Directory: 0x02-functions\_nested\_loops
* File: 4-isalpha.c

 Done? Help Check your code Ask for a new correction Get a sandbox QA Review

5. Sign

**mandatory**

Score: 80.0% (*Checks completed: 80.0%*)

Write a function that prints the sign of a number.

* Prototype: int print\_sign(int n);
* Returns 1 and prints + if n is greater than zero
* Returns 0 and prints 0 if n is zero
* Returns -1 and prints - if n is less than zero

julien@ubuntu:~/0x02$ cat 5-main.c

#include "main.h"

/\*\*

\* main - check the code.

\*

\* Return: Always 0.

\*/

int main(void)

{

int r;

r = print\_sign(98);

\_putchar(',');

\_putchar(' ');

\_putchar(r + '0');

\_putchar('\n');

r = print\_sign(0);

\_putchar(',');

\_putchar(' ');

\_putchar(r + '0');

\_putchar('\n');

r = print\_sign(0xff);

\_putchar(',');

\_putchar(' ');

\_putchar(r + '0');

\_putchar('\n');

r = print\_sign(-1);

\_putchar(',');

\_putchar(' ');

\_putchar(r + '0');

\_putchar('\n');

return (0);

}

julien@ubuntu:~/0x02$ gcc -Wall -pedantic -Werror -Wextra -std=gnu89 \_putchar.c 5-main.c 5-sign.c -o 5-sign

julien@ubuntu:~/0x02$ ./5-sign

+, 1

0, 0

+, 1

-, /

julien@ubuntu:~/0x02$

**Repo:**

* GitHub repository: alx-low\_level\_programming
* Directory: 0x02-functions\_nested\_loops
* File: 5-sign.c

 Done? Help Check your code Ask for a new correction Get a sandbox QA Review

6. There is no such thing as absolute value in this world. You can only estimate what a thing is worth to you

**mandatory**

Score: 80.0% (*Checks completed: 80.0%*)

Write a function that computes the absolute value of an integer.

* Prototype: int \_abs(int);

FYI: The standard library provides a similar function: abs. Run man abs to learn more.

julien@ubuntu:~/0x02$ cat 6-main.c

#include "main.h"

#include <stdio.h>

/\*\*

\* main - check the code

\*

\* Return: Always 0.

\*/

int main(void)

{

int r;

r = \_abs(-1);

printf("%d\n", r);

r = \_abs(0);

printf("%d\n", r);

r = \_abs(1);

printf("%d\n", r);

r = \_abs(-98);

printf("%d\n", r);

return (0);

}

julien@ubuntu:~/0x02$ gcc -Wall -pedantic -Werror -Wextra -std=gnu89 \_putchar.c 6-main.c 6-abs.c -o 6-abs

julien@ubuntu:~/0x02$ ./6-abs

1

0

1

98

julien@ubuntu:~/0x02$

**Repo:**

* GitHub repository: alx-low\_level\_programming
* Directory: 0x02-functions\_nested\_loops
* File: 6-abs.c

 Done? Help Check your code Ask for a new correction Get a sandbox QA Review

7. There are only 3 colors, 10 digits, and 7 notes; it's what we do with them that's important

**mandatory**

Score: 62.5% (*Checks completed: 62.5%*)

Write a function that prints the last digit of a number.

* Prototype: int print\_last\_digit(int);
* Returns the value of the last digit

julien@ubuntu:~/0x02$ cat 7-main.c

#include "main.h"

/\*\*

\* main - check the code

\*

\* Return: Always 0.

\*/

int main(void)

{

int r;

print\_last\_digit(98);

print\_last\_digit(0);

r = print\_last\_digit(-1024);

\_putchar('0' + r);

\_putchar('\n');

return (0);

}

julien@ubuntu:~/0x02$ gcc -Wall -pedantic -Werror -Wextra -std=gnu89 \_putchar.c 7-main.c 7-print\_last\_digit.c -o 7-last\_digit

julien@ubuntu:~/0x02$ ./7-last\_digit

8044

julien@ubuntu:~/0x02$

**Repo:**

* GitHub repository: alx-low\_level\_programming
* Directory: 0x02-functions\_nested\_loops
* File: 7-print\_last\_digit.c

 Done? Help Check your code Ask for a new correction Get a sandbox QA Review

8. I'm federal agent Jack Bauer, and today is the longest day of my life

**mandatory**

Score: 0.0% (*Checks completed: 0.0%*)

Write a function that prints every minute of the day of Jack Bauer, starting from 00:00 to 23:59.

* Prototype: void jack\_bauer(void);
* You can listen to [this soundtrack](https://intranet.alxswe.com/rltoken/aNwRcWg7MPM1J2lYuuuBjA) while coding :)

julien@ubuntu:~/0x02$ cat 8-main.c

#include "main.h"

/\*\*

\* main - check the code

\*

\* Return: Always 0.

\*/

int main(void)

{

jack\_bauer();

return (0);

}

julien@ubuntu:~/0x02$ gcc -Wall -pedantic -Werror -Wextra -std=gnu89 \_putchar.c 8-main.c 8-24\_hours.c -o 8-24

julien@ubuntu:~/0x02$ ./8-24 | head

00:00

00:01

00:02

00:03

00:04

00:05

00:06

00:07

00:08

00:09

julien@ubuntu:~/0x02$ ./8-24 | tail

23:50

23:51

23:52

23:53

23:54

23:55

23:56

23:57

23:58

23:59

julien@ubuntu:~/0x02$ ./8-24 | wc -l

1440

julien@ubuntu:~/0x02$

**Repo:**

* GitHub repository: alx-low\_level\_programming
* Directory: 0x02-functions\_nested\_loops
* File: 8-24\_hours.c

 Done? Help Check your code Ask for a new correction Get a sandbox QA Review

9. Learn your times table

**mandatory**

Score: 0.0% (*Checks completed: 0.0%*)

Write a function that prints the 9 times table, starting with 0.

* Prototype: void times\_table(void);
* Format: see example

julien@ubuntu:~/0x02$ cat 9-main.c

#include "main.h"

/\*\*

\* main - check the code

\*

\* Return: Always 0.

\*/

int main(void)

{

times\_table();

return (0);

}

julien@ubuntu:~/0x02$ gcc -Wall -pedantic -Werror -Wextra -std=gnu89 \_putchar.c 9-main.c 9-times\_table.c -o 9-times\_table

ulien@ubuntu:~/0x02$ ./9-times\_table | cat -e

0, 0, 0, 0, 0, 0, 0, 0, 0, 0$

0, 1, 2, 3, 4, 5, 6, 7, 8, 9$

0, 2, 4, 6, 8, 10, 12, 14, 16, 18$

0, 3, 6, 9, 12, 15, 18, 21, 24, 27$

0, 4, 8, 12, 16, 20, 24, 28, 32, 36$

0, 5, 10, 15, 20, 25, 30, 35, 40, 45$

0, 6, 12, 18, 24, 30, 36, 42, 48, 54$

0, 7, 14, 21, 28, 35, 42, 49, 56, 63$

0, 8, 16, 24, 32, 40, 48, 56, 64, 72$

0, 9, 18, 27, 36, 45, 54, 63, 72, 81$

julien@ubuntu:~/0x02$ ./9-times\_table | tr ' ' . | cat -e

0,..0,..0,..0,..0,..0,..0,..0,..0,..0$

0,..1,..2,..3,..4,..5,..6,..7,..8,..9$

0,..2,..4,..6,..8,.10,.12,.14,.16,.18$

0,..3,..6,..9,.12,.15,.18,.21,.24,.27$

0,..4,..8,.12,.16,.20,.24,.28,.32,.36$

0,..5,.10,.15,.20,.25,.30,.35,.40,.45$

0,..6,.12,.18,.24,.30,.36,.42,.48,.54$

0,..7,.14,.21,.28,.35,.42,.49,.56,.63$

0,..8,.16,.24,.32,.40,.48,.56,.64,.72$

0,..9,.18,.27,.36,.45,.54,.63,.72,.81$

julien@ubuntu:~/0x02$

**Repo:**

* GitHub repository: alx-low\_level\_programming
* Directory: 0x02-functions\_nested\_loops
* File: 9-times\_table.c

 Done? Help Check your code Ask for a new correction Get a sandbox QA Review

10. a + b

**mandatory**

Score: 77.78% (*Checks completed: 77.78%*)

Write a function that adds two integers and returns the result.

* Prototype: int add(int, int);

julien@ubuntu:~/$ cat 10-main.c

#include "main.h"

#include <stdio.h>

/\*\*

\* main - check the code

\*

\* Return: Always 0.

\*/

int main(void)

{

int n;

n = add(89, 9);

printf("%d\n", n);

return (0);

}

julien@ubuntu:~/0x02$ gcc -Wall -pedantic -Werror -Wextra -std=gnu89 \_putchar.c 10-main.c 10-add.c -o 10-add

julien@ubuntu:~/0x02$ ./10-add

98

julien@ubuntu:~/0x02$

**Repo:**

* GitHub repository: alx-low\_level\_programming
* Directory: 0x02-functions\_nested\_loops
* File: 10-add.c

 Done? Help Check your code Ask for a new correction Get a sandbox QA Review

11. 98 Battery Street, the OG

**mandatory**

Score: 0.0% (*Checks completed: 0.0%*)

Write a function that prints all natural numbers from n to 98, followed by a new line.

* Prototype: void print\_to\_98(int n);
* Numbers must be separated by a comma, followed by a space
* Numbers should be printed in order
* The first printed number should be the number passed to your function
* The last printed number should be 98
* You are allowed to use the standard library

julien@ubuntu:~/0x02$ cat 11-main.c

#include "main.h"

/\*\*

\* main - check the code

\*

\* Return: Always 0.

\*/

int main(void)

{

print\_to\_98(0);

print\_to\_98(98);

print\_to\_98(111);

print\_to\_98(81);

print\_to\_98(-10);

return (0);

}

julien@ubuntu:~/0x02$ gcc -Wall -pedantic -Werror -Wextra -std=gnu89 \_putchar.c 11-main.c 11-print\_to\_98.c -o 11-98

julien@ubuntu:~/0x02$ ./11-98

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98

98

111, 110, 109, 108, 107, 106, 105, 104, 103, 102, 101, 100, 99, 98

81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98

-10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98

julien@ubuntu:~/0x02$

**Repo:**

* GitHub repository: alx-low\_level\_programming
* Directory: 0x02-functions\_nested\_loops
* File: 11-print\_to\_98.c

 Done? Help Check your code Ask for a new correction