Learning Chat Gitea Pr Pending 6640 QPoints azimjan_bo

Quest02

Track Bootcamp C Arc 01

Subject Additional Resources (8)

Quest02

Remember to git add && git commit && git push each exercise!

We will execute your function with our test(s), please DO NOT PROVIDE ANY TEST(S) in your file

For each exercise, you will have to create a folder and in this folder, you will have additional files that contain your work. Folder names are provided at the beginning of each exercise under `submit directory` and specific file names for each exercise are also provided at the beginning of each exercise under `submit file(s)` .

Pointers!!

In our life as a developer, there is a before and after with pointers :D
There is no better way to learn pointers than by doing it.

What is a pointer?

```
A variable which contain the address of another variable.
```

^o)
Let dive... in?

Control Center

Also working on the project

njoroge_p abdoul-k_r issa-him_d mahamado_h
yeshmura_j abdumal_a timirkha_t azikhanu_c
balykbai_a deribsal_a ziyanuro_z mardenov_l
komiljon_k christi_nk

Just finished

rempanot_c ross_s naimanba_a abduvakh_b
kalauov_t skemetr_t abitov_a kumar_ro
dadwal_p

The objective of this quest is to discover pointers and start to get experience with pointers.
We will also do some loops. :)

Quest02	My Initializer
Submit directory	ex00
Submit file	my_initializer.c

Description

Create a function that takes a pointer to integer as a parameter, and sets the value to `0` .

Function prototype (c)

```
/*
**
** QWASAR.IO -- my_initializer
**
** @param {int*} param_1
**
** @return {void}
**
*/

void my_initializer(int* param_1)
{

}
```

Example 00 (In C)

Type	Project
Group Size	1 Participant
Review system	Test Review (Gandalf)
Difficulty	Initiation
Average duration	1 Week

Project's Metadata

Track	Project
id: 1200	id: 32
name: Bootcamp C Arc 01	name: quest02
visible: True	visible: True

```c
int main() {
  int variable_a = 12;

  printf("%d\n", variable_a); // will print 12
  my_initializer(&variable_a);
  printf("%d\n", variable_a); // will print 0
  return 0;
}
```

Quest02	My Swap
Submit directory	ex01
Submit file	my_swap.c

Description

Let's switch the contents of parameter A and parameter B. :-)

Create a function that swaps the value of two integers whose addresses are entered as parameters.

Function prototype (c)

```c
/*
**
** QWASAR.IO -- my_swap
**
** @param {int*} param_1
** @param {int*} param_2
**
** @return {void}
**
*/

void my_swap(int* param_1, int* param_2)
{

}
```

Example 00 (In C)

```
int main() {
  int variable_a = 12;
  int variable_b = 21;

  printf("A - %d ** B - %d\n", variable_a, variable_b); // will print A - 12 ** B - 21
  my_swap(&variable_a, &variable_b);
  printf("A - %d ** B - %d\n", variable_a, variable_b); // will print A - 21 ** B - 12
  return 0;
}
```

Tip
(In C)
Pointer is a key element here. Remember a pointer is a variable that contains the address of another variable.

Quest02	My Strlen
Submit directory	ex02
Submit file	my_strlen.c

Description

Reproduce the behavior of the function strlen.
The strlen() function computes the length of the string s.

The strlen() function returns the number of characters.
C Prototype:

```
size_t my_strlen(const char *s);
```

Example 00:

```
Input: "abc"
Output: 3
```

Example 01:

```
Input: "RaInB0w d4Sh!"
Output: 13
```

Example 02:

```
Input: "ThE C4k3 Is a L|3"
Output: 17
```

Function prototype (c)

```
/*
**
** QWASAR.IO -- my_strlen
**
** @param {char*} param_1
**
** @return {int}
**
*/

int my_strlen(char* param_1)
{

}
```

Quest02	My Putstr
Submit directory	ex03
Submit file	my_putstr.c

Description

Create a function that displays a string of characters on the standard output. The address of the string's first character is in the pointer entered as parameter in the function.

Function prototype (c)

```c
/*
**
** QWASAR.IO -- my_putstr
**
** @param {char*} param_1
**
** @return {void}
**
*/

void my_putstr(char* param_1)
{

}
```

Example 00

```
Input: "abc"
Output: abc
Return Value: nil
```

Example 01

```
Input: "abcdelele dzp ll 0"
Output: abcdelele dzp ll 0
Return Value: nil
```

Example 02

```
Input: ""
Output:
Return Value: nil
```

Tips
(In C)
Remember `\0` is the End Of String
(In C)
To print a character you can use `my_putchar`

```
int my_putchar(char c) {
  return write(1, &c, 1);
}
```

Quest02	My Add
Submit directory	ex04
Submit file	my_add.c

Description

Create a `my_add` function which takes `2` parameters (`nbr1` and `nbr2`) and returns a `value` .
This `value` is the result of the addition of `nbr1` and `nbr2` parameters.

Function prototype (c)

```
/*
**
** QWASAR.IO -- my_add
**
** @param {int} param_1
** @param {int} param_2
**
** @return {int}
**
*/

int my_add(int param_1, int param_2)
{

}
```

Example 00

```
Input: 0 && 1
Output:
Return Value: 1
```

Example 01

```
Input: 10 && 10
Output:
Return Value: 20
```

Example 02

```
Input: -10 && 10
Output:
Return Value: 0
```

Quest02	My Sub
Submit directory	ex05
Submit file	my_sub.c

Description

Create a `my_sub` function which takes `2` parameters (`nbr1` and `nbr2`) and returns a `value` .
This `value` is the result of the subtraction of `nbr1` and `nbr2` parameters.

Function prototype (c)

```c
/*
**
** QWASAR.IO -- my_sub
**
** @param {int} param_1
** @param {int} param_2
**
** @return {int}
**
*/

int my_sub(int param_1, int param_2)
{

}
```

Example 00

```
Input: 0 && 1
Output:
Return Value: -1
```

Example 01

```
Input: 10 && 10
Output:
Return Value: 0
```

Example 02

```
Input: -10 && 10
Output:
Return Value: -20
```

Quest02	My Mult
Submit directory	ex06

Submit file	my_mult.c

Description

Create a my_mult function which takes 2 parameters (nbr1 and nbr2) and returns a value .
This value is the result of the multiplication of nbr1 and nbr2 parameters.

Function prototype (c)

```c
/*
**
** QWASAR.IO -- my_mult
**
** @param {int} param_1
** @param {int} param_2
**
** @return {int}
**
*/

int my_mult(int param_1, int param_2)
{

}
```

Example 00

```
Input: 0 && 1
Output:
Return Value: 0
```

Example 01

```
Input: 10 && 10
Output:
Return Value: 100
```

Example 02

```
Input: -10 && 10
Output:
Return Value: -100
```

Quest02	My String Formatting
Submit directory	ex07
Submit file	my_string_formatting.c

Description

Create a `my_string_formatting` function which takes `3` parameters (`firstname` , `lastname` and `age`) and prints a string composed `value` .

Formatting should be: "Hello, my name is FIRSTNAME LASTNAME, I'm AGE."

Make sure you are printing a newline.

Function prototype (c)

```
/*
**
** QWASAR.IO -- my_string_formatting
**
** @param {char*} param_1
** @param {char*} param_2
** @param {int} param_3
**
** @return {void}
**
*/

void my_string_formatting(char* param_1, char* param_2, int param_3)
{

}
```

Example 00

```
Input: "john" && "doe" && 37
Output: Hello, my name is john doe, I'm 37.

Return Value: nil
```

Example 01

```
Input: "Baby" && "Yoda" && 50
Output: Hello, my name is Baby Yoda, I'm 50.

Return Value: nil
```

Example 02

```
Input: "Marie" && "Curie" && 26
Output: Hello, my name is Marie Curie, I'm 26.

Return Value: nil
```

Tip
You should use Google to learn about String interpolation :-)

Quest02	My String Index
Submit directory	ex08
Submit file	my_string_index.c

Description

Create a `my_string_index` function which takes `2` parameters (`haystack` and `needle`) and locates the first occurrence of the character needle in the string haystack and returns the position.

You can think of this function as: is there an L (character) in my string "helLo"?

The objective is to build a loop that has an if statement which returns the characters position when it matches the `right` character.

Function prototype (c)

```
/*
**
** QWASAR.IO -- my_string_index
**
** @param {char*} param_1
** @param {char} param_2
**
** @return {int}
**
*/

int my_string_index(char* param_1, char param_2)
{

}
```

Example 00

```
Input: "hello" && "l"
Output:
Return Value: 2
```

Example 01

```
Input: "aaaaa" && "b"
Output:
Return Value: -1
```

Quest02	My Upcase
Submit directory	ex09
Submit file	my_upcase.c

Description

Create a `my_upcase` function that takes a string as a parameter and returns the uppercase version of it.

Function prototype (c)

```
/*
**
** QWASAR.IO -- my_upcase
**
** @param {char*} param_1
**
** @return {char*}
**
*/

char* my_upcase(char* param_1)
{

}
```

Example 00

```
Input: "aBc"
Output:
Return Value: "ABC"
```

Example 01

```
Input: ""
Output:
Return Value: ""
```

Tips
Google upcase string YOURCODINGLANGUAGE
(In C)

```
/*
Example of main
*/
int main() {
  char *my_str = strdup("AbcE Fgef1");

  printf("RANDOM CASE -> %s\n", my_str);
  printf("UPCASE      -> %s\n", my_upcase(my_str));
  return 0;
}
```

Quest02	My Downcase
Submit directory	ex10
Submit file	my_downcase.c

Description

Create a `my_downcase` function that takes a string as a parameter and returns the lowercase version of it.

Function prototype (c)

```
/*
**
** QWASAR.IO -- my_downcase
**
** @param {char*} param_1
**
** @return {char*}
**
*/

char* my_downcase(char* param_1)
{

}
```

Example 00

```
Input: "aBc"
Output:
Return Value: "abc"
```

Example 01

```
Input: ""
Output:
Return Value: ""
```

Tips
Google downcase string YOURCODINGLANGUAGE
(In C)

```
/*
Example of main
*/
int main() {
  char *my_str = strdup("AbcE Fgef1");

  printf("RANDOM CASE -> %s\n", my_str);
  printf("DOWNCASE    -> %s\n", my_downcase(my_str));
  return 0;
}
```

Qwasar - Terms of Service - Web Accessibility - Privacy Policy