Assignment 5

Exercise 00: vc_str_is_alpha

Turn-in files	vc_str_is_alpha.c
Allowed functions	Nothing

- Create a function that returns 1 if the string given as a parameter contains only alphabetical characters, and 0 if it contains any other character. It should return 1 if **str** is empty.
- Function prototype:
  - `int vc_str_is_alpha(char *str);`

Exercise 01: vc_str_is_numeric

Turn-in files	vc_str_is_alpha.c
Allowed functions	Nothing

- Create a function that returns 1 if the string given as a parameter contains only digits, and 0 if it contains any other character. It should return 1 if **str** is empty.
- Function prototype:
  - `int vc_str_is_numeric(char *str);`

Exercise 02: vc_str_is_lowercase

Turn-in files	vc_str_is_lowercase.c
Allowed functions	Nothing

- Create a function that returns 1 if the string given as a parameter contains only lowercase alphabetical characters, and 0 if it contains any other character. It should return 1 if **str** is empty.
- Function prototype:
  - `int vc_str_is_lowercase(char *str);`

Exercise 03: vc_str_is_uppercase

Turn-in files	vc_str_is_uppercase.c
Allowed functions	Nothing

- Create a function that returns 1 if the string given as a parameter contains only uppercase alphabetical characters, and 0 if it contains any other character. It should return 1 if **str** is empty.

- Function prototype:
    - `int vc_str_is_uppercase(char *str);`

Exercise 04: vc_str_is_printable

Turn-in files	**vc_str_is_printable.c**
Allowed functions	Nothing

- Create a function that returns 1 if the string given as a parameter contains only printable characters, and 0 if it contains any other character. It should return 1 if **str** is empty.
- Function prototype:
    - `int vc_str_is_printable(char *str);`

Exercise 05: vc_strcat

Turn-in files	**vc_strcat.c**
Allowed functions	Nothing

- Reproduce the behavior of the function **strcat**.
- Reference: `man strcat`
- Function prototype:
    - `char *vc_strcat(char *dest, char *src);`

Exercise 06: vc_strncat

Turn-in files	**vc_strncat.c**
Allowed functions	Nothing

- Reproduce the behavior of the function **strncat**.
- Reference: `man strncat`
- Function prototype:
    - `char *vc_strncat(char *dest, char *src, int n);`

Exercise 07: vc_strlcat

Turn-in files	**vc_strlcat.c**
Allowed functions	Nothing

- Reproduce the behavior of the function **strlcat**.

- Reference: `man strlcat`
- Function prototype:
  - `unsigned int vc_strlcat(char *dest, char *src, unsigned int size);`

Exercise 08: vc_strlcpy

Turn-in files	vc_strlcpy.c
Allowed functions	Nothing

- Reproduce the behavior of the function **strlcpy**.
- Reference: `man strlcpy`
- Function prototype:
  - `unsigned int *vc_strlcpy(char *dest, char *src, unsigned int size);`

Exercise 09: vc_putstr_non_printable

Turn-in files	vc_putstr_non_printable.c
Allowed functions	putchar

- Create a function that displays a string of characters onscreen. If this string contains characters that aren't printable, they'll have to be displayed in the shape of hexadecimals (lowercase), preceeded by a "backslash".
- Hint: ASCII table 0 ~ 31 are not printable.
- For example:
  - `Hello\nwhat is your favorite food?`
- Becomes:
  - `Hello\0awhat is your favorite food?`
- Function prototype:
  - `void vc_putstr_non_printable(char *str);`

Exercise 10: vc_print_memory

Turn-in files	vc_print_memory.c
Allowed functions	putchar

- Create a function that displays the memory area onscrean.
- The display of this memory area should be split into three columns :

    - The hexadecimal address of the first line's first character;
    - The content in hexadecimal
    - The content in printable characters.

- If a character is non-printable, it will be replaced by a dot.
- Each line should handle sixteen characters.
- If the `size` equals to 0, nothing should be displayed.
- It should return **addr**
- For example:

```
# Test string (char *)
"Salut les aninches c'est cool show non on fait de truc
terrible\x00\x2e\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0e\x0f\x1b\x7f"
```

```
dp@ciccc $ ./vc_print_memory
00000000: 5361 6c75 7420 6c65 7320 616d 696e 6368 Salut les aminch
00000010: 6573 2063 2765 7374 2063 6f6f 6c20 7368 es c'est cool sh
00000020: 6f77 206d 656d 206f 6e20 6661 6974 2064 ow mem on fait d
00000030: 6520 7472 7563 2074 6572 7269 626c 6500 e truc terrible.
00000040: 2e00 0102 0304 0506 0708 090e 0f1b 7f ..............

dp@ciccc $ ./vc_print_memory | cat -te
00000000: 5361 6c75 7420 6c65 7320 616d 696e 6368 Salut les aminch$
00000010: 6573 2063 2765 7374 2063 6f6f 6c20 7368 es c'est cool sh$
00000020: 6f77 206d 656d 206f 6e20 6661 6974 2064 ow mem on fait d$
00000030: 6520 7472 7563 2074 6572 7269 626c 6500 e truc terrible.$
00000040: 2e00 0102 0304 0506 0708 090e 0f1b 7f ..............$
```

- Function prototype: `void ft_print_memory(void *addr, unsigned int size);`