Assignment 2

Exercise 00: vc_pt

Turn-in files	vc_pt.c
Allowed functions	Nothing

- Create a function that takes a pointer to int as a parameter, and sets the value "77" to that **int**.
- Function prototype: `void vc_pt(int *n);`

Exercise 01: vc_ultimate_pt

Turn-in files	vc_ultimate_pt.c
Allowed functions	Nothing

- Create a function that takes a pointer to pointer to pointer to pointer to pointer to pointer to pointer to pointer to pointer to int as a parameter and sets the value "77" to that int.
- Function prototype: `void vc_ultimate_pt(int *********n);`

Exercise 02: vc_swap

Turn-in files	vc_swap.c
Allowed functions	Nothing

- Create a function that swaps the value of two integers whose addresses are entered as parameters.
- Function prototype: `void vc_swap(int *a, int *b);`

Exercise 03: vc_div_mod

Turn-in files	vc_div_mod.c
Allowed functions	Nothing

- Create a function that divides parameters **a** by **b** and stores the result in the *int* pointed by **div** . It also stores the remainder of the division of **a** by **b** in the **int** pointed by **mod**.
- Function prototype: `void vc_div_mod(int a, int b, int *div, int *mod);`

Exercise 04: vc_ultimate_div_mod

Turn-in files	vc_ultimate_div_mod.c
Allowed functions	Nothing

- Create a function that divides parameters **a** by **b**. The result of this division is stored in the int pointed by **a**. The remainder of the division is stored in the int pointed by **b**.
- Function prototype: `void vc_ultimate_div_mod(int *a, int *b);`

Exercise 05: vc_putstr

Turn-in files	**vc_putstr.c**
Allowed functions	putchar

- Create a function that displays a string of characters on the standard output.
- Function prototype: `void vc_putstr(char *str);`

Exercise 06: vc_strlen

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

- Create a function that counts and returns the number of characters in a string.
- Function prototype: `int vc_strlen(char *str);`

Exercise 07: vc_strrev

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

- Create a function that reverses the order of characters in a string.
- It has to return **str**.
- For example:

```
a => a
ab => ba
abcde => edcba
```

- Function prototype: `char *vc_strrev(char *str);`

Exercise 08: vc_atoi

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

- Reproduce the behavior of the function **atoi**

- Reference: `man atoi`
- Function prototype: `int vc_atoi(char *str);`

Exercise 09: vc_sort_int_table

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

- Create a function which sorts an array(table) of integers by ascending order.
- The arguments are a pointer to ***int*** and teh number of ints in the array.
- Function prototype: `void vc_sort_int_table(int *tab, int size);`