Programming Assignment 6

In the **Programming/Programming06** folder of your class repository on GitHub, create a file named **util.h** with a header guard. Only include: **iostream, string, cmath,** and **stdexcept .** All tasks should be in a namespace called **dshw**.

Task 1: (0.5 points)

Initialize a constant string variable with the characters listed in the table of Task 2.

Task 2: (1 point)

Create a function called **getValue()** that takes a character parameter. The function should return the decimal value of the character parameter. Use the table below for the proper conversions. If the character is not a valid character, return -1. Letters are case insensitive.

char	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'	'A'	'B'	'C'	'D'	'E'	'F'
int	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Task 3: (1.5 points)

Create a function called **validate()** that takes a constant string reference parameter and an integer parameter. The string parameter represents a signed binary, octal, or hexadecimal number. The integer parameter represents the base of the number. Return true if every character in the string is valid for the base. 2: (0,1), 8: (0-7), 16: (0-9, A-F, a-f)

> Example: **validate("0123", 2) => false, validate("1010", 2) => true**
>
> **validate("678", 8) => false, validate("71450", 8) => true**
>
> **validate("3BALL", 16) => false, validate("ACE5", 16) => true**

Task 4: (3 points)

Create a function called **pad()** that takes a constant string reference parameter and two integer parameters. The string parameter represents a signed binary, octal, or hexadecimal number. The second integer parameter represents the base of the number. If the first integer parameter is greater than the length of the string parameter, return a padded version of the string number whose length equals the first integer parameter. The padded value should be that of the sign digit. Ensure the string is valid for the base. If the string is invalid, throw an exception.

> Example: **pad("1010", 8, 2) => "11111010", pad("010", 5, 2) => "00010"**
>
> **pad("6010", 6, 8) => "776010", pad("334", 6, 8) => "000334"**
>
> **pad("CAFE", 5, 16) => "FCAFE", pad("7D5", 6, 16) => "0007D5"**

Task 5: (4 points)

Create a function called **trim()** that takes a constant string reference parameter and an integer parameter. The string parameter represents a signed binary, octal, or hexadecimal number. The integer parameter represents the base of the number. Return a string that represents the string number with any padded sign digits removed. Ensure the string is valid for the base. If the string is invalid, throw an exception.

> Example: **trim("11111010", 2) => "1010", trim("00000111", 2) => "0111"**
>
> **trim("7770765", 8) => "70765", trim("0000655", 8) => "0655"**
>
> **trim("7775765", 8) => "5765", trim("0002655", 8) => "2655"**
>
> **trim("FFF4CE5", 16) => "F4CE5", trim("0000A", 16) => "0A"**
>
> **trim("FFFACE5", 16) => "ACE5", trim("00067", 16) => "67"**