



DASF004: Basic and Practice in Programming

❖ Lab 5: Function2



In this lab ...

- ❖ Arguments Coercion
- ❖ Recursive Functions
- ❖ Random Function
- ❖ What you need to submit in this lab (Lab #5):
 - » Lab Exercise 5 by the end of today (Wednesday 11:59 pm)
 - » Assignment #5 by Tuesday 11:59 pm



Argument Coercion

When the variable(s) passing as argument(s) to the function is in a different type as defined by the function, the variable will be cast into the type defined by the function before passing as arguments(s).

```
float MyFunction(float input);    // function prototype
... ..
int input = 32;
float result = MyFunction(input); // function call
```

The integer `input` will be casted into float, and then pass to `MyFunction`



Try to write a function to convert temperature degree Celsius to Fahrenheit.

For example: $0\text{ }^{\circ}\text{C} = 32\text{ }^{\circ}\text{F}$

The formula for the conversion is:

$$T(^{\circ}\text{F}) = T(^{\circ}\text{C}) \times 9/5 + 32$$



Try it yourself

```
#include <stdio.h>
```

```
float Fahrenheit2Celsius(float input);
```

```
int main(void)
{ float Celsius1 = 32.0;
  int Celsius2 = 28;
  printf("%f\n", Fahrenheit2Celsius(Celsius1));
  printf("%f\n", Fahrenheit2Celsius(Celsius2));
  return 0;
}
```

```
float Fahrenheit2Celsius(float input);
{ // Write your function here!
}
```

D:\PortableApp\Dev-Cpp32\ConsolePauser.exe

90.500000

82.400000

Process exited normally.

Press any key to continue . . .

A Recursive Power Function

- Recursive function invokes/calls itself
 - Successive calls reduce to simpler task
 - Until base case with trivial solution reached
- The n^{th} power of a number

Iteratively

$$r^n = \underbrace{r \cdot r \cdot \dots \cdot r}_{n \text{ terms}}$$

Recursively

$$\begin{aligned} r^1 &= r \\ r^n &= r \cdot r^{n-1} \end{aligned}$$

A Recursive Power Function

Traits of recursive algorithms

1. One or more base cases with direct solutions.

2. An "inductive step"

- Reducing the problem to one or more smaller versions of the same problem
- Reduction eventually culminating in a base case.
- Called the reducing step.

A Recursive Power Function

```
float power(float x, float n)
{ if (n == 1)
    return x;
  else
    return x * power(x,n-1);
}
```


Lab Exercise #5: Recursive Function



The task of this assignment is to write a program to find the Greatest Common Denominator of two integers using recursion.

Your program should ask user to enter two integers.

The recursive algorithm is as follow (Also known as Euclid's Algorithm):
Given two positive integers, m and n

1. If the value of m is less than the value of n, switch the values of m and n
2. If n equals to 0, result = m and quit (base case)
3. If n does not equal to 0, go to step 4
4. $m = n$
5. $n = m \% n$
6. Goto Step 2 (recursive case)

Lab Exercise #5: Recursive Function

Sample outputs:

M = 102, N = 38

GCD = 2

Process exited after 0.01336 seconds with return value 0

Press any key to continue . . .

M = 6, N = 3

GCD = 3

Process exited after 0.01327 seconds with return value 0

Press any key to continue . . .

M = 24, N = 128

GCD = 8

Process exited after 0.01341 seconds with return value 0

Press any key to continue . . .

M = 103, N = 4

GCD = 1

Process exited after 0.01321 seconds with return value 0

Press any key to continue . . .



Random Number

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main(void)
{ srand(time(NULL)); // generate random seed

  int randomNumber = rand() % 6 + 1; // random number range [1,6]
  printf("RandomNumber: %d\n", randomNumber);
  return 0;
}
```

Assignment #5: Random Number



Create a program that will simulate the casino game of Sic Bo

https://en.wikipedia.org/wiki/Sic_bo

Your program will roll 3 dices, and display the result of the 3 dices
It will also display one of the three result of the Sic Bo game based on the 3 dices:

1. If it is a Triple (1 1 1; 2 2 2; 3 3 3; 4 4 4; 5 5 5 or 6 6 6), display "TRIPLE!"
2. Else if the sum is 3 to 10, display "SMALL!".
3. Else if the sum is 11 to 17, display "BIG!".



Sample output

 C:\Users\Arthur Tang\Documents\Untitled1.exe

```
Result: 5 6 5  
BIG!  
  
-----  
Process exited after 0.01221 seconds with return value 0  
Press any key to continue . . .
```

 C:\Users\Arthur Tang\Documents\Untitled1.exe

```
Result: 4 1 2  
SMALL!  
  
-----  
Process exited after 0.01336 seconds with return value 0  
Press any key to continue . . .
```

Lab Assignment #5: Function

Submit your source code on iCampus before Tuesday 11:59 pm