Subject: PRF192- PFC
Workshop 04: FUNCTION

Objectives:

- Practicing skills at analyzing and implementing the complex functions
- Designing and implementing menu program with multiple choice.

Program 1 (3.5 marks)

Objectives	Practice implementing a program with complex function
Related knowledge	A number is called as **Strong** number when it is a satisfying number whose sum of the factorials of its digits is equal to itself. Example: 145 = 1! + 4! + 5!
Problem	Write a program that inputs a number n with no more than 6 digits and lists the Strong numbers less than n (Satisfactory results are listed on a single line separated by a space)

Analysis
Nouns: positive long integer → int n
	Suggested algorithm (logical order of verbs)
	Char IsStrong (int n) {
	tong=0,check;
	check=n;
	while (n>=1) {
	int giaiThua = 1;
	if (n Chia 10 =0 hoac 1) {
	tong ++;
	n = n chia 10 lay nguyen;
	continue;
	}
	For(i=1 to n%10) Do {
	giaiThua *= i;
	}
	tong += giaiThua;
	a /= 10;
	}
	if (tong==check) return YES;
	else return NO;
	}
	Begin
	Do {
	Accept n;
	} While (n<1 || n>1000000);
	For (i = 1 to n) Do {
	If (IsStrong(i)) Print i

	} End

Program 2 (3.5 marks)

Objectives	Practice implementing a program with simple menu
Related knowledge	None
Problem	Write a C program that will execute repetitively using a simple menu 1- Process primes 2- Process strong; 3- Quit Select an operation: 1- When user selects the option 1, the program will accept a positive integral number and print out a message about whether the input number is a prime or not. 2- When user selects the option 2, the program will accept a positive integral number and print out a message about whether the input number is a Strong number or not. 3- The program will terminate when user selects the option 3.

Analysis
Nouns:
- positive integral number → int n
- A number represents a choice of user → int choice;
Functions: int prime(int n) → see previous problem
Functions: int strong(int n) → see previous problem
	Suggested algorithm (logical order of verbs)
	Begin
	Do /* Print out the menu and get user choice*/
	{
	Print out "1- Process primes\n";
	Print out "2- Process strong\n";
	Print out "3- Quit\n";
	Print out "Select an operation:";
	switch(choice) {
	case 1: do {
	Input n;
	} while(n<0);
	If (prime(n)==1) Print " It is a prime\n";
	Else Print " It is not a prime\n";
	break;
	case 2: do {
	Input n;
	} while(n<0);
	If (strong(n)==1) Print " It is a strong\n";
	Else Print " It is not a strong\n";
	break;

	} while (choice >0 & choice<3); End

Program 3 (3 marks)

FINANCIAL CALCULATOR

Design and code a program that performs two financial calculations: future value and present value. Your program prompts for and accepts a principal amount, an interest rate, the number of periods and the type of calculation requested: future or present value.

Design your program according to structured design principles and include a function that can be used in both calculations as well as in other applications. Do not use any library functions apart from **<stdio.h>** functions.

Preface your function header with a comprehensive description of the function purpose, the function parameters and the function return value.

The formula for future value is

$$\text{future value} = \text{principal} * (1 + \text{rate})^{(\text{number of periods})}$$

The formula for present value is

$$\text{present value} = \text{principal} * (1 + \text{rate})^{-(\text{number of periods})}$$

The output from your program should look something like:

```
Investment Calculator
=====================
Principal   :  1000
Annual Rate:  0.06
No of Years:  5
Future value (f) or present value (p): f

The present amount : $  1000.00
The future value   : $  1338.23

Investment Calculator
=====================
Principal   :  1000
Annual Rate:  0.06
No of Years:  5
```

```
Future value (f) or present value (p): p

The future amount  : $  1000.00
The present value  : $   747.26
```