2024 NCKU Program Design I Homework 1

> 請不要嘗試攻擊 Judge 系統，否則此堂課將不予以通過
>
> Don't attack judge system otherwise you will fail this course.
>
> 一旦發現作業抄襲或是請人代寫之情形，學期作業成績將 "全部" 以 0 分計算
>
> One instances of severe plagiarism, hiring someone to write assignments, or similar activities are detected, the semester's assignment scores will be calculated as 0 point across the board.

> 對於作業有任何問題可以直接寄信到 f74114744@gs.ncku.edu.tw (mailto:f74114744@gs.ncku.edu.tw)，如果是其他課程上的問題需要處理請利用全體助教信箱 c2024@mail.csie.ncku.edu.tw (mailto:c2024@mail.csie.ncku.edu.tw)
>
> If you have any question about this homework tasks (ex. problem description), please feel free to contact me (資訊 115 陳俊安, f74114744@gs.ncku.edu.tw (mailto:f74114744@gs.ncku.edu.tw)).
>
> Otherwise, the common problem of this homework please send to TAs mail (c2024@mail.csie.ncku.edu.tw (mailto:c2024@mail.csie.ncku.edu.tw))

> 2024/09/14 輸出完後請記得換行，否則在 Server 上會被判定是錯誤的

Homework 1 Information

Deadline: 09/25 23:59:59

> 不接受遲交
>
> Late submissions are not accepted.

Before you start

- Make sure you can login the server by your personal account.

Goals

- Basic C language Input and Output
- Variable and Type

- Expression为

Grading (Total: 100 points)

Task	Score	Tags
A	20	Input & Output
B	20	Input Format Handling
C	20	Expression
D	40	Expression

Submission

- Server IP: `140.116.246.48` , Port: `2024`

```
ssh 學號@140.116.246.48 -p 2024
```

- Login the system by your personal account. (Use the ssh command)

> 登入後請記得更改自己的密碼
>
> Please remember to change your password after logging in.

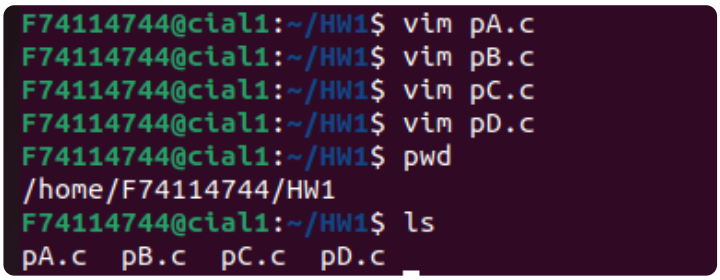
- Create an directory with name "HW1" in your home directory.
  - You can use the "pwd" command to confirm your current directory.

```
F74114744@cial1:~$ pwd
/home/F74114744
F74114744@cial1:~$
```

  - The "mkdir [name]" command can create a directory with the name [name]

```
F74114744@cial1:~$ mkdir HW1
F74114744@cial1:~$ ls
HW1
```

- In HW1 directory, you need to create 4 files with name "pA.c", "pB.c", "pC.c" and "pD.c" respectively.



- You can directly use the command `hw1` to check whether the result of each question is correct.

Tasks

Problem A. Welcome to NCKU Program Design I (20%)

First, let's start by trying basic input and output.

Please design a program that takes a positive integer N as input and outputs 'Welcome to NCKU Program Design I N'

▶ If you have never learned this before, you can click here to see a simple explanation of scanf and printf.

Input Format

```
N
```

- $2011 \leq N \leq 2024$

> This means that our test data will not input numbers greater than 2024 and will not small than 2011, so you don't need to write a specific 'if' statement to check this range.

Sample Input 1

```
2024
```

Sample Output 1

```
Welcome to NCKU Program Design I 2024
```

Sample Input 2

```
2011
```

Sample Output 2

```
Welcome to NCKU Program Design I 2011
```

Problem B. Scanf Data Extraction (20%)

The `scanf` function in C has a special use. It can handle input formatting in a unique way. Let's look at the example below.

```c
#include <stdio.h>
int main() {
    int x;
    scanf("NCKU%dCSIE",&x);
    printf("%d",x);
}
```

In this way, `scanf` will understand that the desired input format is `NCKU[x (int)]CSIE`, so when I input `NCKU2024CSIE`, `x` will be correctly read as the value `2024` and output accordingly.

In this exercise, please practice this functionality.

The input format is `NCKU CSIE[x (int)]Programing Design I.`
Please output the value of `x` hidden inside.

Sample Input 1

```
NCKU CSIE2024Programing Design I.
```

Sample Output 1

```
2024
```

Sample Input 2

```
NCKU CSIE2011Programing Design I.
```

Sample Output 2

```
2011
```

Problem C. Comparison Expression (20%)

A comparison expression returns `true` or `false` depending on whether the given statement is valid. Here, `true` represents `1`, while `false` represents `0`.

Let's take an example:

```
#include <stdio.h>
int main() {
    int n;
    scanf("%d",&n);
    printf("%d",(n == 10));
}
```

When we input `10`, the output will be `1`; otherwise, the output will be `0`.

`==` is used to check whether the left and right sides are equal. You can try using these comparison operators yourself. Other common operators used for comparison include `>`, `>=`, `<`, `<=`, `!=`, and so on.

```
#include <stdio.h>
int main() {
    int a,b;
    scanf("%d %d",&a,&b);
    printf("%d\n",(a == b));
    printf("%d\n",(a >= b));
    printf("%d\n",(a <= b));
    printf("%d\n",(a != b));
    printf("%d\n",(a < b));
    printf("%d\n",(a > b));
}
```

Next, let's practice this. Please design a program that takes two positive integers as input and outputs whether the first integer is not equal to the second. If they are not equal, output `1`; otherwise, output `0`.

Input Format

- $1 \leq a, b \leq 10^9$

```
a b
```

Sample Input 1

```
10 20
```

Sample Output 1

```
1
```

Sample Input 2

```
100 100
```

Sample Output 2

```
0
```

Problem D. A Simple Calculator (40%)

Next, let's combine what we've learned from the previous three exercises and implement a simple calculator.

Please design a program that takes a simple mathematical expression as input and calculates the correct result. To simplify the task, you only need to handle addition (`+`), subtraction (`-`), and multiplication (`*`).

> Although it's not mandatory, there is a way to solve this problem without using `if/else` .
>
> If you have learned about this, you can think about it.

▼ Hint 1

Please observe the output of the following program.

```c
#include <stdio.h>
int main() {
    char c = '-';
    printf("%d\n",(c == '+'));
    printf("%d\n",(c == '-'));
    printf("%d\n",(c == '*'));
}
```

▼ Hint 2

Make use of the characteristic where an expression that is `true` becomes `1` , and an expression that is `false` becomes `0` .

▼ Hint 3

You can create an expression that calculates all three results and only multiply one part by `1`, while multiplying the others by `0`.

▼ Hint 4

if input `10-20`:

$(10 + 20) \times 0 + (10 - 20) \times 1 + (10 * 20) \times 0 = -10$

Input Format

- $1 \leq a, b \leq 100$
- It is guaranteed that the characters will only be `+`, `-`, `*`.

```
a[+|-|*]b
```

Sample Input 1

```
10+20
```

Sample Output 1

```
30
```

Sample Input 2

```
10-20
```

Sample Output 2

```
-10
```

Sample Input 3

```
10*20
```

Sample Output 3

```
200
```