

Lab sheet 6 – Loop statements

รหัสนิสิต ชื่อ-สกุล หมู่ปฏิบัติการที่

1. โปรแกรมต่อไปนี้คำนวณค่าแฟคตอเรียล (factorial) ของตัวเลขจำนวนเต็ม n จากสูตรดังนี้

$$n! = n * (n-1) * (n-2) * \dots * 2 * 1;$$

ให้นักเติมส่วนของโปรแกรมนี้ให้สมบูรณ์ ให้สามารถคำนวณค่าแฟคตอเรียลได้ เช่น $5! = 5*4*3*2*1$

```
#include <stdio.h>
void main() {
    int i;                      // Loop counter
    int n;                      // Number whose factorial is to be computed
    long factorial = 1;         // Factorial initialized
    printf("Enter number: ");
    scanf("%d",&n);

    for (__(a)__;__(b)__;__(c)__) // Loop from n to 1.
        factorial = factorial * __(d)__;

    printf("%d! is %ld\n",n,factorial);
}
```

(a)	(b)	(c)	(d)

- 1.1 หลังจากเติมส่วนของโปรแกรม และทดลองรันโปรแกรมข้างต้นแล้ว ให้นักทดลองป้อนค่า n ต่าง ๆ แล้วให้นักอธิบายการทำงานของโปรแกรม

- 1.2 เขียนโปรแกรมใหม่โดยใช้ while statement ให้สามารถแสดงผลลัพธ์แบบเดียวกัน

2. โปรแกรมต่อไปนี้แสดงตัวเลข 5, 10, ..., 5k โดยที่ k ได้มาจากการป้อนค่าทางคีย์บอร์ด จงเติมคำสั่งที่หายไป

```
int i, k;
scanf("%d",&k);

for (__(a)__;__(b)__;__(c)__)
    printf("%d ",i);
```

(a)	(b)	(c)

3. จงเขียนโปรแกรมเพื่อแสดงค่า 2^1 ถึง 2^{10} ให้ได้ผลลัพธ์ดังตัวอย่าง

```
===Power of 2===
2^1 is 2
2^2 is 4
2^3 is 8
2^4 is 16
2^5 is 32
2^6 is 64
2^7 is 128
2^8 is 256
2^9 is 512
2^10 is 1024
```

4. ให้แก้ไขโปรแกรมในข้อ 3 เพื่อแสดงค่า 2^{10} ถึง 2^{50} ให้ได้ผลลัพธ์ดังตัวอย่าง

```
Kilobyte to Petabyte (Number of bytes)
2^10 bytes is 1024 bytes
2^20 bytes is 1048576 bytes
2^30 bytes is 1073741824 bytes
2^40 bytes is 1099511627776 bytes
2^50 bytes is 1125899906842624 bytes
```

5. จงเขียนโปรแกรมเพื่อรับจำนวนเต็มบวก n แล้วพิมพ์ตัวเลขคู่ที่อยู่ระหว่าง 0 ถึง n ออกมาทางหน้าจอ หากผู้ใช้ป้อนค่า n ที่ไม่ใช่จำนวนเต็มบวก ให้ผู้ใช้ป้อนค่าใหม่จนกว่าจะป้อนจำนวนเต็มบวก

ตัวอย่างที่ 1 Enter n: 7 Even number 0-7 is 2 4 6	ตัวอย่างที่ 3 Enter n: -11 Enter n: 0 Enter n: 14 Even number 0-14 is 2 4 6 8 10 12 14
ตัวอย่างที่ 2 Enter n: 11 Even number 0-11 is 2 4 6 8 10	

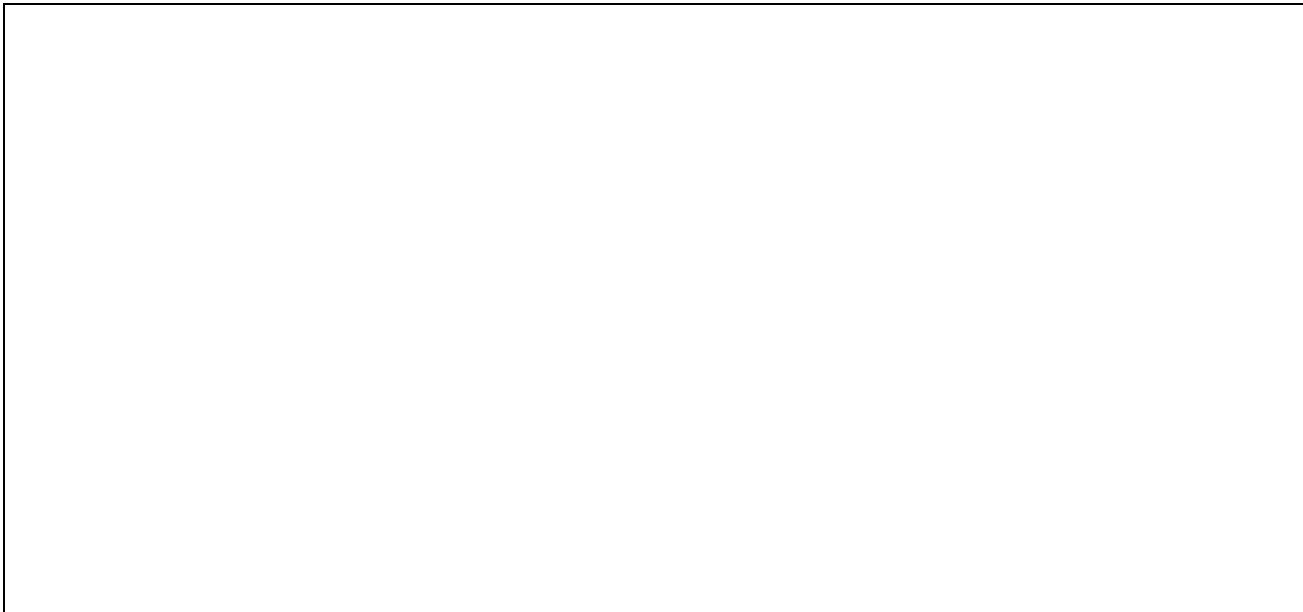
6. จงเขียนโปรแกรมเพื่อรับจำนวนเต็ม a แล้วหาจำนวนเต็มบวก b ที่มีค่าน้อยที่สุดและมีค่ามากกว่า 1 ที่หาร a ลงตัว (*hint: ให้เขียนคำสั่งวนซ้ำเพื่อตรวจสอบค่าของตัวแปร b หาก b หาร a ไม่ลงตัว ให้เพิ่มค่า b ขึ้น 1 แล้ววนซ้ำ*)

ตัวอย่างที่ 1 Enter a: 7 Value of b is 7	
ตัวอย่างที่ 2 Enter a: 8 Value of b is 2	
ตัวอย่างที่ 3 Enter a: 15 Value of b is 3	

7. เขียนโปรแกรมเพื่อแสดงรายงานยอดเงินบัญชีเงินฝากของ Mary หลังดำเนินการฝาก (Deposit)-ถอน (Withdraw) เงินตามจำนวนเงินที่ต้องการ โดยเมื่อเลือกตัวเลือกที่ 1 ให้ดำเนินการฝากเงิน เมื่อเลือกตัวเลือกที่ 2 ให้ดำเนินการถอนเงิน และเมื่อเลือกตัวเลือกที่ 3 ให้ออกจากโปรแกรม (กำหนดให้มียอดเงินในบัญชีเริ่มต้น 1000.59 US dollar)

ตัวอย่าง (ตัวเอียงหนา คือค่าที่ป้อนทางคีย์บอร์ด)

<p>Hello Mary, what would you like to do? (1) Deposit (2) Withdraw (3) Exit Please select:1 How much your amount? 1222</p> <p>DUMMY REPORT: Mary had \$1000.59 Mary deposited +1222.00 Now she has \$2222.59</p> <p>(1) Deposit (2) Withdraw (3) Exit Please select:2 How much your amount? 2000</p> <p>DUMMY REPORT: Mary had \$2222.59 Mary withdrew -2000.00 Now she has \$222.59</p> <p>(1) Deposit (2) Withdraw (3) Exit Please select:3 Bye, Mary</p>	<p>Hello Mary, what would you like to do? (1) Deposit (2) Withdraw (3) Exit Please select:2 How much your amount? 100</p> <p>DUMMY REPORT: Mary had \$1000.59 Mary withdrew -100.00 Now she has \$900.59</p> <p>(1) Deposit (2) Withdraw (3) Exit Please select:3 Bye, Mary</p>
---	--



8. (Optional) จงเขียนโปรแกรมทายตัวเลขที่ได้จากการสุ่มตัวเลขระหว่าง 0-32767 โดยโปรแกรมจะรับค่าเลขจำนวนเต็มจากคีย์บอร์ดเรื่อยๆ จนกว่าตัวเลขที่รับมานั้นมีค่าตรงกันกับตัวเลขที่ทาย

เพื่อให้การทายง่ายขึ้น หากตัวเลขที่รับมามีค่าน้อยกว่าตัวเลขที่ทายให้แสดงข้อความว่า “Too low” พร้อมทั้งกำหนดค่าเริ่มต้นของช่วงที่ต้องทายค่าใหม่ หากตัวเลขที่รับมามีค่ามากกว่าตัวเลขที่ทายให้แสดงข้อความว่า “Too high” พร้อมกำหนดค่าสิ้นสุดของช่วงที่ต้องทายค่าใหม่

ผลการทำงานแสดงดังตัวอย่างที่ 1-3 โดยตัวเลขที่ได้จากการสุ่ม คือ 4086, 849, 32213 ตามลำดับ

ตัวอย่างที่ 1 Guessing number between 0-32767: 4000 Too low, Try again Guessing number between 4000-32767: 4086 Good guess, well done!	ตัวอย่างที่ 3 Guessing number between 0-32767: 30000 Too low, Try again Guessing number between 30000-32767: 32750 Too high, Try again Guessing number between 30000-32750: 32000 Too low, Try again Guessing number between 32000-32750: 32500 Too high, Try again Guessing number between 32000-32500: 32300 Too high, Try again Guessing number between 32000-32300: 32100 Too low, Try again Guessing number between 32100-32300: 32200 Too low, Try again Guessing number between 32200-32300: 32213 Good guess, well done!
ตัวอย่างที่ 2 Guessing number between 0-32767: 500 Too low, Try again Guessing number between 500-32767: 1000 Too high, Try again Guessing number between 500-1000: 900 Too high, Try again Guessing number between 500-900: 800 Too low, Try again Guessing number between 800-900: 850 Too high, Try again Guessing number between 800-850: 845 Too low, Try again Guessing number between 845-850: 849 Good guess, well done!	

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int ans,n;
    int start=0,end=RAND_MAX;           //RAND_MAX is a constant
    srand((unsigned) time(NULL));       //Initializes random number generator
    ans=rand();                          // Function for generating random number between 0- RAND_MAX
    /*
        Loop for guessing number
    */
}
```