

## Quiz 4-A: Functions

คะแนนรวม

20/20



ระบบบันทึกอีเมลของผู้ตอบ (thitichot.k@ku.th) ไว้แล้วเมื่อส่งแบบฟอร์มนี้

0 จาก 0 คะแนน

รหัสนิสิต \*

6610402001

คำปฏิญาณ "ข้าพเจ้าจะทำแบบทดสอบนี้โดยปฏิบัติตามคำชี้แจงของอาจารย์ผู้สอน ทำด้วย \*  
ความซื่อสัตย์ ทำด้วยตนเอง และจะไม่คัดลอก ไม่ปรึกษา ไม่ถามคำตอบและไม่ตอบคำถาม  
ของผู้อื่น รวมถึงไม่ให้ผู้อื่นทำแทน และหากพบว่าข้าพเจ้าทุจริตในการทดสอบครั้งนี้  
ข้าพเจ้ารับทราบบทลงโทษและข้าพเจ้าจะถูกดำเนินการตามข้อบังคับของทางมหาวิทยาลัย  
ที่ได้กำหนดเอาไว้"

พิมพ์ชื่อและนามสกุลของตนเอง

ฐิติโชติ กิ่งไพบูลย์

นิสิตต้องตอบคำถามให้กระจ่าง และหากคำตอบมีความคลุมเครือ คำตอบนั้นจะไม่ได้ 20 จาก 20  
คะแนน ในกรณีที่คำถามมีตัวเลือกมาให้ นิสิตต้องเลือกทุกตัวเลือกที่มีส่วนถูกต้อง 20 คะแนน



ให้นิยามฟังก์ชันที่มีชื่อว่า product โดยฟังก์ชันนี้รับพารามิเตอร์ 2 ตัวที่เป็น x และ y ซึ่งมีประเภทข้อมูลเป็น float และ int ตามลำดับ นอกจากนี้ฟังก์ชัน product ส่งค่าที่มีประเภทข้อมูลเป็นเลขจำนวนจริง 32 บิตกลับไปยังผู้เรียกโดยเป็นผลคูณของ x และ y จงเลือกตัวเลือกที่เหมาะสมสำหรับแต่ละช่องหมายเลขในรูป

```
1 2 ( 3 4 , 5 6 )
{
  7 x * y;
}
```

	def	int	float	double	return	x	product	y	pass
หมายเลข 1 และ 2	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
หมายเลข 3 และ 4	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
หมายเลข 5 และ 6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
หมายเลข 7	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

หากนำค่า NULL มาเปรียบเทียบกับค่าในตัวเลือก NULL จะมีค่าเท่ากับตัวเลือกใดบ้าง 1/1

- ☐ EOF
- ☐ 1
- ☒ '\0'
- ☐ ไม่มีข้อใดถูก
- ☒ 0
- ☐ -1

```
#include <stdio.h>

void swap(int x, int y)
{ int temp;

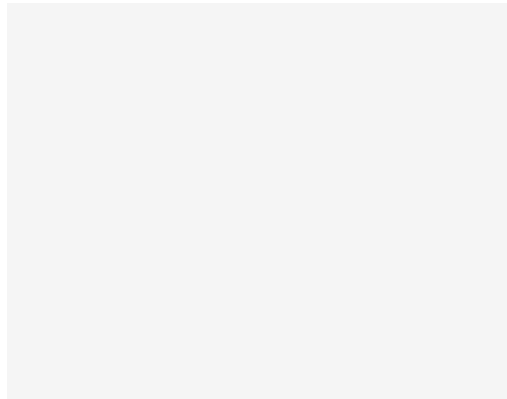
  temp = x;
  x = y;
  y = temp;
}

int main()
{ int x = 3, y = 4;

  printf("ก่อน x = %d, y = %d\n", x, y);
  swap(x, y);
  printf("หลัง x = %d, y = %d\n", x, y);
}
```

- ☐ ก่อน x = 4, y = 3
- ☒ หลัง x = 3, y = 4
- ☐ ไม่มีข้อใดถูก
- ☐ หลัง x = 4, y = 3
- ☒ ก่อน x = 3, y = 4

ฟังก์ชันต่อไปนี้จะส่งค่ากลับเป็นผลรวมของค่าข้อมูลในอาร์เรย์ที่ส่งผ่านเข้ามาในฟังก์ชัน 1/1  
ได้อย่างถูกต้อง โดยค่าพารามิเตอร์ n บอกถึงจำนวนข้อมูลที่อยู่ในอาร์เรย์



☐ ไม่มีข้อใดถูก

```
float sum(float data[], int n)
{ int i, total=0;

  for (i=0; i < n; i++)
    total = total + data[i];
  return total;
}
```

☐ 31851064

```
float sum(float data[], int n)
{ int i;
  float total=0;

  for (i=0; i < n; i++)
    total = total + data[i];
  return sum;
}
```

☐ 38151064

```
int sum(int data[], int n)
{ int i, total=0;

  for (i=0; i < n; i++)
    total = total + data[i];
  return total;
}
```

☒ 31850164

```
double sum(float data[], int n)
{ int i;
  double total=0;

  for (i=0; i < n; i++)
    total = total + data[i];
  return total;
}
```

☒ 38115064

## Stack Frames เก็บข้อมูลอะไรบ้าง

1/1

- ☒ ตัวแปร local ที่ถูกประกาศอยู่ในฟังก์ชัน
- ☒ ตัวแปรพารามิเตอร์ที่ส่งผ่านเข้ามาในฟังก์ชัน
- ☐ ตำแหน่งในหน่วยความจำของฟังก์ชันที่ต้องการเรียกใช้
- ☐ ไม่มีข้อใดถูก

## โปรแกรมนี้คอมไพล์ผ่านหรือไม่ หากคอมไพล์ไม่ผ่าน อะไรคือสาเหตุ

1/1

```
#include <stdio.h>

int main()
{ int cnt;

    for (cnt=1; cnt <= 10; cnt++) {
        int sum=0;

        sum += cnt;
    }
    printf("sum = %d\n", sum);
}
```

- ☐ คอมไพล์ผ่านแต่มี warning
- ☐ คอมไพล์ไม่ผ่าน เนื่องจากตัวแปร cnt ไม่สามารถถูกเข้าถึงภายในลูป for ได้
- ☒ คอมไพล์ไม่ผ่าน เนื่องจากฟังก์ชัน printf พยายามพิมพ์ค่า sum ซึ่งถูกประกาศไว้ภายในลูป for
- ☐ คอมไพล์ผ่านโดยไม่มีปัญหาแต่อย่างใด
- ☐ ไม่มีข้อใดถูก



```
1 #include <stdio.h>
2
3 int f1(int i, int j)
4 {
5     return i+j;
6 }
7
8 int main()
9 { int x=10;
10
11     printf("f1(10, 20) = %d\n", f1(2*x, x+20));
12 }
```

- ☐ Call by Name
- ☒ Call by Value
- ☐ Call by Function
- ☐ ไม่มีข้อใดถูก
- ☐ Call by Expression



ด้านล่างนี้เป็นโค้ดของฟังก์ชัน `get_line` ที่ใช้รับสตริง 1 บรรทัดมาจากคีย์บอร์ด จงกำหนด function header ของฟังก์ชันนี้ให้เหมาะสมโดยพารามิเตอร์ตัวแรกของฟังก์ชันเป็นอาร์เรย์ที่ใช้เก็บค่าสตริง ส่วนพารามิเตอร์อีกตัวเป็นจำนวนไบต์สูงสุดที่สามารถจัดเก็บสตริงลงในอาร์เรย์นี้ได้ และฟังก์ชันนี้ส่งกลับจำนวนอักขระที่ถูกพิมพ์ได้

```

1 2 ( 3 , 4 )
{ int i;

    for (i=0; i < size-1 && (str[i]=getchar()) != '\n'; i++) ;
    str[i] = 0;
    return i;
}

```

void char int str str[]; size; str[] size get\_line

หมายเลข 1 และ 2	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
หมายเลข 3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
หมายเลข 4	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

```
1 #include <stdio.h>
2
3 int f1(int, int);
4
5 int main()
6 {   int f2();
7
8     printf("f1(10, 20) = %d\n", f1(10, 20));
9     printf("f2(10, 20) = %d\n", f2(10, 20));
10 }
11
12 int f1(int i, int j)
13 {
14     return i+j;
15 }
16
17 int f2(i, j)
18     int i, j; // old-style C parameter declarations
19 {
20     return i+j;
21 }
```

- ☐ 8
- ☐ 17
- ☐ 14
- ☐ 1
- ☐ ไม่มีข้อใดถูก
- ☒ 6
- ☐ 9
- ☐ 18
- ☐ 5
- ☐ 20
- ☒ 3
- ☐ 12



หากเราต้องการใช้ฟังก์ชัน `log()` เราควร `#include` ไฟล์อะไรเข้ามาในโปรแกรม เพื่อให้ 1/1 เวลาคอมไพล์แล้วไม่มี warning เกิดขึ้น

- ☐ `stdlib.h`
- ☐ `log.h`
- ☐ ไม่มีข้อใดถูก
- ☒ `math.h`
- ☐ `unistd.h`
- ☐ `stdio.h`

ฟังก์ชันต่อไปนี้นี้เป็นฟังก์ชันที่มีการเรียกซ้ำ (Recursion) ที่ส่งค่ากลับเป็นผลรวมของค่า 1/1 ข้อมูลในอาร์เรย์ที่ส่งผ่านเข้ามาในฟังก์ชัน โดยค่าพารามิเตอร์ `n` บอกถึงจำนวนข้อมูลที่อยู่ในอาร์เรย์

```
int sum(int data[], int n)
{
    int total=0;

    if (n == 0)
        return data[n];
    else
        return n + sum(data, n-1);
}
```

☐ 38115064

☐ ไม่มีข้อใดถูก

```
int sum(int data[], int n)
{
    int total=0;

    if (n == 0)
        return 0;
    else
        return sum(data, n-1) + data[n-1];
}
```

☒ 31851064

```
int sum(int data[], int n)
{
    int total=0;

    if (n == 0)
        return 0;
    else
        return data[n-1] + sum(data, n-1);
}
```

☒ 31850164



การคอมไพล์ด้วย option -l ใน gcc เพื่อให้โปรแกรมสามารถเรียกใช้ฟังก์ชันในไลบรารี static ได้นั้น ทำได้อย่างไร 1/1

- ☒ เพื่อบอกคอมไพเลอร์ให้นำ object code ของฟังก์ชันที่ต้องการเรียกใช้จากไลบรารีที่ระบุมาลิงก์ร่วมกับ object code ที่เรียกใช้ฟังก์ชัน
- ☒ ปกติบนระบบ 64 บิต หาก gcc ถูกเรียกด้วย -lfunc การลิงก์จะหาฟังก์ชันที่อยู่ในไฟล์ libfunc.a รวมถึงฟังก์ชันที่อยู่ใน libc.a ที่อยู่ในโฟลเดอร์ /usr/lib64 เพื่อลิงก์เข้ามา
- ☐ เพื่อบอกคอมไพเลอร์ให้นำซอร์สโค้ดของฟังก์ชันที่ต้องการเรียกใช้มาคอมไพล์
- ☐ เพื่อบอกคอมไพเลอร์ให้นำ object code ของฟังก์ชันทั้งหมดในไลบรารีที่ต้องการมาลิงก์ร่วมกับ object code ที่เรียกใช้ฟังก์ชัน

ตัวเลือกใดกล่าวไม่ถูกต้องเกี่ยวกับการใช้ #include แล้วตามด้วยชื่อไฟล์ header ใน ภาษา C 1/1

- ☒ เราสามารถใช้ตัวแปรสตริงในการกำหนดชื่อไฟล์ header ได้
- ☒ โดยทั่วไปแล้ว #include "ชื่อไฟล์ header" จะนำไฟล์ header มาจากโฟลเดอร์ /usr/include
- ☐ เราสามารถใช้ #include <ชื่อไฟล์ header> และ #include "ชื่อไฟล์ header" ในการนำไฟล์เข้ามาในโค้ด
- ☒ ไฟล์ header ของไลบรารีฟังก์ชัน C Standard จะเก็บซอร์สโค้ดของฟังก์ชันต่าง ๆ เช่น printf เอาไว้

โปรแกรมนี้จะแสดงอะไรขึ้นมานบนหน้าจอ หลังจากที่ป้อนข้อมูลที่กำหนดไว้ให้ในด้านล่าง 2/2 ของโค้ดหมดแล้ว

```
#include <stdio.h>

unsigned char zero(int d)
{   int c;

    if ((c=getchar()) == EOF)
        return d;
    else
        return zero(d + (c >= '0' && c <= '9')) + (c == '0');
}

int main()
{
    printf("%d", zero(0));
}
```

ข้อมูลข้างล่างนี้เป็นข้อมูลที่จะถูกป้อนเข้ามาในโปรแกรม โดยหลังจากพิมพ์ข้อมูลครบหมดแล้ว ผู้ป้อนกด Ctrl-d เพื่อแจ้งกับโปรแกรมว่าหมดข้อมูลที่ต้องการป้อนแล้ว

68  
33  
28  
80  
79  
28  
73  
46  
80  
12  
92  
30  
42  
73  
70

ตัวเลือกใดบ้างที่กล่าวถูกต้องเกี่ยวกับค่า return address

1/1

- ☐ ไม่มีข้อใดถูก
- ☒ เป็นตำแหน่งของคำสั่งที่ต้องทำงานถัดไปหลังจากที่ฟังก์ชันที่ถูกเรียกทำงานเสร็จสิ้น
- ☒ return address จะถูกเก็บขึ้นเมื่อมีการเรียกใช้ฟังก์ชัน
- ☐ เป็นค่าที่เก็บตำแหน่งของผลลัพธ์ที่จะถูกส่งกลับมาจากฟังก์ชัน หลังการเรียกใช้ฟังก์ชัน
- ☒ เป็นค่าที่ถูกเก็บอยู่ใน stack frame
- ☐ เป็นค่าที่ถูกส่งกลับมาจากฟังก์ชัน หลังการเรียกใช้ฟังก์ชัน

แบบฟอร์มนี้ถูกสร้างขึ้นภายใน มหาวิทยาลัยเกษตรศาสตร์

Google ฟอร์ม



