

Quiz 6-A: Pointers

thitichot.k@ku.th [สลับบัญชี](#)



ส่งอีกครั้งเพื่อบันทึก

ระบบจะบันทึกอีเมลของคุณเมื่อส่งแบบฟอร์มนี้

นิสิตต้องตอบคำถามให้กระจ่าง และหากคำตอบมีความคลุมเครือ คำตอบนั้นจะไม่ได้รับคะแนน ในกรณีที่คำถามมีตัวเลือกมาให้ นิสิตต้องเลือกทุกตัวเลือกที่มีส่วนถูกต้อง

ฟังก์ชัน average ทำหน้าที่หาค่าเฉลี่ยจากข้อมูลที่อยู่ในอาร์เรย์ data โดยจะส่งค่าเฉลี่ยที่คำนวณได้กลับไปยังผู้เรียกผ่านตัวแปรพารามิเตอร์ (บรรทัดที่ 14 แสดงการเรียกใช้ฟังก์ชัน average) จงเลือกตัวเลือกที่เหมาะสมให้กับส่วนของ #A#, #B#, #C#, และ #D# เพื่อให้ฟังก์ชันนี้ทำงานถูกต้อง

```
1 void average(int data[], int size, #A#)
2 { int i=0, sum=0;
3
4     for (#B#)
5         #C#
6
7     #D#
8 }
9
10 int main()
11 { int data[] = {1, 2, 3, 4};
12   float value;
13
14   average(data, 4, &value);
15 }
```

- ☐ #A#: int return;
- ☐ #A#: int *return_value;
- ☒ #A#: float *return_value;
- ☐ #A#: double *return_value;
- ☒ #B#: i=0; i < size; i++
- ☐ #B#: i=1; i <= size; i++
- ☐ #B#: i=0; i < size-1; i++
- ☐ #C#: sum += data;
- ☐ #C#: sum = sum + *data;
- ☒ #C#: sum = sum + data[i];
- ☐ #D#: return_value = sum / size;
- ☐ #D#: *return_value = sum / size;
- ☐ #D#: return_value = sum / (float)size;
- ☒ #D#: *return_value = sum / (float)size;

อะไรบ้างที่สามารถเป็นสาเหตุให้โปรแกรมเกิด segmentation fault

- ☐ ไม่มีข้อใดถูก
- ☒ การใช้ค่า index กับอาร์เรย์ที่เกินขนาดของอาร์เรย์
- ☒ โปรแกรมพยายามที่จะอ่านค่าในหน่วยความจำที่ไม่มีสิทธิ์ในการเข้าถึง
- ☒ โปรแกรมพยายามที่จะเขียนค่าในหน่วยความจำที่ไม่มีสิทธิ์ในการเข้าถึง

หากโปรแกรมนีได้รับตัวเลขจำนวนเต็ม 2 จำนวนที่มีค่าเป็น -2 และ 1 เข้ามาในโปรแกรม จงบอกผลลัพธ์การทำงานของโปรแกรมนี

```
#include <stdio.h>

void action(double (*pfunc)(int, float), int x, float y)
{
    printf("%.2lf\n", (*pfunc)(x, y));
}

double f1(int x, float y)
{
    return x + y;
}

double f2(int x, float y)
{
    return x * y;
}

int main ()
{
    int x, y;
    double (*f)(int, float);

    scanf("%d %d", &x, &y);
    f = (x + y) ? f1 : f2;
    action(f, 2, 2.5);
}
```

4.50

ตัวดำเนินการ (operator) ไหนบ้างที่สามารถใช้โดยตรงกับตัวแปร pointer ได้

- ☐ - (unary operator)
- ☐ & (binary operator)
- ☒ & (unary operator)
- ☐ * (binary operator)
- ☒ - (binary operator)
- ☒ * (unary operator)
- ☒ + (binary operator)

การเรียกใช้ h("hello") จะแสดงข้อความใดออกมาบนหน้าจอ

```
void h(char a[])
{
    while (*a != '\0') {
        printf("%c", *a-1);
        a++;
    }
}
```

- ☐ ไม่มีข้อใดถูก
- ☐ fcjjm
- ☒ gdkkn
- ☐ HELLO
- ☐ ifmmp
- ☐ hello

จงบอกผลลัพธ์ที่จะปรากฏบนหน้าจอจากการทำงานของโปรแกรมนี้

```
#include <stdio.h>

int main()
{   int d[] = {1, 2, 3, 4};
    int *p;

    for (p=d; *p != 6; p = d + (++p - d) % 4)
        *p = *p * 2;
    printf("%d %d %d %d", d[0], d[1], d[2], d[3]);
}
```

4 8 6 8

หลังจากฟังก์ชัน swap ถูกเรียกใช้ในบรรทัดที่ 14 ค่าที่อยู่ในตัวแปร x และ y จะมีค่าเป็นเท่าไร

```
1  #include <stdio.h>
2
3  void swap(int *x, int *y)
4  {   int temp = *x;
5
6      *x = *y;
7      *y = temp;
8  }
9
10 int main()
11 {   int x = 10;
12     int y = 20;
13
14     swap(&x, &y);
15 }
```

- ☒ 20, 10
- ☐ 10, 10
- ☐ 10, 20
- ☐ 30, 30
- ☐ 20, 20

การประกาศตัวแปรใดที่มีชนิดเป็น function pointer

- ☐ int *f1();
- ☐ ไม่มีข้อใดถูก
- ☒ int (*f2)();
- ☒ int *(*f3)();

จงบอกผลการรันของโปรแกรมนี้

```
#include <stdio.h>

int main()
{ char s1[] = {'u', 'n', 'i', 'v', 'e', 'r', 's', 'i', 't', 'y'};
  char s2[] = "university";

  printf("%d", sizeof(s1)-sizeof(s2));
}
```

-1

จากโค้ดด้านล่างนี้ เมื่อโปรแกรมทำงานจนถึงบรรทัดที่ 7 และสามารถจองพื้นที่ในหน่วยความจำได้ จงตอบคำถามเกี่ยวกับ memory segment ดังต่อไปนี้

```
1 #include <stdio.h>
2
3 int main()
4 { int *p;
5
6     p = (int *)malloc(400);
7     getchar();
8 }
```

	text segment	initialized data segment	uninitialized data segment	heap segment	stack segment
segment ของหน่วยความจำที่ชี้จัดเก็บตัวแปร p	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
segment ของหน่วยความจำที่ตัวแปร p อ้างอิงไป	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

ตัวเลือกใดเป็นจริงที่เกี่ยวกับฟังก์ชัน malloc() และ free()

- ☒ free() มีไว้คืนพื้นที่หน่วยความจำที่เคยได้จองไว้ คืนให้กับระบบ
- ☒ malloc() มีไว้จองพื้นที่ในหน่วยความจำ
- ☐ ไม่มีข้อใดถูก
- ☒ พารามิเตอร์ที่ malloc() รับเข้าไปในฟังก์ชันเป็นจำนวนไบต์ที่ต้องการจองพื้นที่
- ☐ หาก malloc() ไม่สามารถจองพื้นที่ในหน่วยความจำได้ โปรแกรมจะจบการทำงาน

กำหนดให้การทำงานภายในโปรแกรมเป็นดังรูป หลังจากโค้ดนี้ทำงานเสร็จ จงบอกค่าของนิพจน์ $*(b+1)$ และ $*c$

```
int *a, *b, *c;

a = (int *)malloc(6 * sizeof(int));

// หลังจากนั้นมีการกำหนดค่าให้กับ a[0] = 10, a[1] = 14, a[2] = 18,
// a[3] = 22, a[4] = 26, a[5] = 30

b = a + 1;
c = b + 4;
```

- ☐ 11, 14
- ☐ ค่าข้อมูลที่ตัวแปร c อ้างอิงไปมีค่าไม่แน่นอน เนื่องจาก c อ้างอิงเกินกว่าหน่วยความจำที่ได้จองไว้
- ☐ 15, 26
- ☒ 18, 30
- ☐ 12, 14

หากโปรแกรมนี้อักรันบนระบบปฏิบัติการ 64 บิต จงบอกผลลัพธ์ที่จะปรากฏบนหน้าจอ

```
#include <stdio.h>

int main()
{ char c, *cp;
  float f, *fp;

  printf("%d %d %d %d", sizeof(c), sizeof(cp),
        sizeof(f), sizeof(fp));
}
```

1 8 4 8

จงบอกผลลัพธ์ที่จะปรากฏบนหน้าจอจากการทำงานของโปรแกรมนี้

```
#include <stdio.h>

int main()
{ char x[] = {2, 8, 10, 9, 5, 7, -1, 6, 3, 4, 1};
  char *p;

  for (p=x; *p != -1; p = x + *p)
    printf("%d ", *p);
}
```

2 10 1 8 3 9 4 5 7 6

กลับ

ส่ง

หน้า 2 จาก 2

ห้ามส่งรหัสผ่านใน Google ฟอรัม

แบบฟอรัมนี้ถูกสร้างขึ้นภายใน มหาวิทยาลัยเกษตรศาสตร์ [รายงานการละเมิด](#)

Google ฟอรัม

