

Quiz 5-A: Program Structure

pongsiri.g@ku.th สลับบัญชี

 ส่งอีกครั้งเพื่อบันทึก

ระบบจะบันทึกอีเมลของคุณเมื่อส่งแบบฟอร์มนี้

นิสิตต้องตอบคำถามให้กระจ่าง และหากคำตอบมีความคลุมเคลือ คำตอบนั้นจะไม่ได้รับคะแนน ในกรณีที่คำถามมีตัวเลือกมาให้ นิสิตต้องเลือกทุกตัวเลือกที่มีส่วนถูกต้อง

การประกาศตัวแปร static โดยไม่ได้กำหนดค่าเริ่มต้นให้กับตัวแปร ตัวแปรนี้จะมีค่าเริ่ม 1 คะแนน
ต้นในตัวแปรเป็นอย่างไร

- ☐ ค่าไม่แน่นอน ซึ่งขึ้นอยู่กับว่าค่าในหน่วยความจำที่ค้างอยู่จะมีค่าเป็นเท่าใด
- ☐ ตัวแปรจะยังไม่ถูกเก็บในหน่วยความจำ จนกว่าจะมีการกำหนดค่าให้กับตัวแปร
- ☒ 0 เสมอ
- ☐ ไม่มีข้อใดถูก

ตัวแปรในโค้ดด้านล่างนี้จะถูกเก็บไว้ใน memory segment ไດ

3 คะแนน

```
int a, b=1;
int main()
{ auto char c=1;
  static short d, e=1;

}
```

	a	b	c	d	e
Heap Segment	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Initialized Data Segment	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Text Segment	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Stack Segment	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Uninitialized Data Segment	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

หลังจากที่โปรแกรมเริ่มทำงาน หากฟังก์ชัน func ถูกเรียกใช้ บรรทัดที่ 9 จะพิมพ์ข้อความอย่างไรออกมา

1 คะแนน

```
1  #include <stdio.h>
2
3  int value = 10;
4
5  void func()
6  { static int value;
7
8      value = value;
9      printf("value = %d\n", value);
10 }
```

- ☒ พิมพ์ว่า value = 0
- ☐ ค่าของ value ที่ถูกพิมพ์ออกมาเป็นเลขจำนวนเต็มซึ่งมีค่าไม่แน่นอน
- ☐ พิมพ์ว่า value = 10
- ☐ ไม่มีข้อใดถูก

ล้างสิ่งที่เลือก

พวกเราคิดว่าผลลัพธ์ของการรันโปรแกรมนี้เป็นอย่างไร

1 คะแนน

```
#include <stdio.h>
int x;
int main()
{
    for (x=0; x < 5; x++)
        printf("%d, ", x);
    printf("%d, ", x);
}
```

0, 1, 2, 3, 4, 0,

กำหนดให้ตัวแปร w ถูกประกาศดังนี้

1 คะแนน

char w[] = {96, 94, 91, 86, 81, 75, 72, 66, 60, 58, 46, 43, 39, 32, 27, 13};

จงบอกค่าที่จะถูกส่งกลับมาหลังจากการเรียก

bar(w, 46, 0, 15)

```
int bar(char buf[], int x, int a, int b)
{
    int c;

    if (a > b)
        return -1;
    c = (a + b) / 2;
    if (x == buf[c])
        return c;
    else
        return (x > buf[c]) ? bar(buf, x, a, c-1) : bar(buf, x, c+1, b);
}
```

10

โค้ดนี้จะแสดงดอกจัน (*) ออกมากี่อัน

1 คะแนน

```
int main()
{  short buf[] = {21, 23, 24, 25, 26, 27, 28, 29, 30, 31,
                  32, 35, 36, 37, 38, 39, 42, 46, 47, 48};
  int l, m=0, n=19;
  do {
    l = m;
    l += n;
    l /= 2;
    if (buf[l] == 44)
      break;
    else if (buf[l] > 44)
      n = --l;
    else
      m = ++l;
    putchar('*');
  } while (m <= n);
}
```

5

จงบอกความแตกต่างระหว่าง static library และ shared library

1 คะแนน

- ☒ การคอมไพล์โปรแกรมด้วย static library จะสร้างไฟล์ไบนารีที่รวมเอาโค้ดที่เรียกใช้ฟังก์ชันและฟังก์ชันที่ถูกเรียกใช้เข้าด้วยกัน
- ☒ การใช้ shared library ช่วยลดการใช้พื้นที่ในหน่วยความจำ โดยฟังก์ชันในไลบรารีที่ถูกโหลดไว้แล้ว สามารถให้หลายโปรแกรมเรียกใช้งานได้ โดยไม่จำเป็นต้องโหลดเข้ามาอีก
- ☐ ขนาดไฟล์ไบนารีที่คอมไพล์ด้วย shared library มักจะมีขนาดใหญ่กว่าคอมไพล์ด้วย static library
- ☐ การสร้างไฟล์ไบนารีของโปรแกรมด้วย static library หากไลบรารีมีการแก้ไข ไฟล์ไบนารีไม่จำเป็นต้องถูกคอมไพล์ใหม่

ข้อใดกล่าวถูกต้องเกี่ยวกับ Dynamic Storage Duration

1 คะแนน

- ☒ สามารถใช้ฟังก์ชัน malloc() ในการจองหน่วยความจำ
- ☐ เป็นการจองหน่วยความจำให้กับโปรแกรมก่อนโปรแกรมเริ่มทำงาน
- ☐ เป็นการจองหน่วยความจำที่ไม่จะต้องระบุขนาดการจอง
- ☒ เป็นการจองหน่วยความจำเพิ่มหลังจากที่โปรแกรมเริ่มทำงานไปแล้ว

หน่วยความจำที่ถูกจองเพิ่มขณะที่โปรแกรมกำลังทำงานอยู่ (dynamic memory allocation) จะถูกจัดเก็บในหน่วยความจำส่วนใด

1 คะแนน

- ☐ data segment
- ☐ text segment
- ☒ heap segment
- ☐ stack segment
- ☐ ไม่มีข้อใดถูก

ล้างสิ่งที่เลือก

หลังจากโค้ดในรูปทำงานคำสั่งในบรรทัดที่ 13 เสร็จเรียบร้อยแล้ว ค่าในตัวแปร cnt จะมีค่าเป็นเท่าไร 1 คะแนน

```
1 int main()
2 {
3     char ans='y';
4     int cnt, score=0;
5
6     cnt = 0;
7     if (ans == 'y') {
8         int score=1;
9     }
10    else {
11        int score=0;
12    }
13    cnt = cnt + score;
14 }
```

0

หลังจากที่โปรแกรมเริ่มทำงาน หากฟังก์ชัน func ถูกเรียกใช้ บรรทัดที่ 9 จะพิมพ์ข้อความอย่างไรออกมา

1 คะแนน

```
1  #include <stdio.h>
2
3  int value = 10;
4
5  void func()
6  {  extern int value;
7
8     value = value;
9     printf("value = %d\n", value);
10 }
```

- ☐ พิมพ์ว่า value = 0
- ☐ ค่าของ value ที่ถูกพิมพ์ออกมาเป็นเลขจำนวนเต็มซึ่งมีค่าไม่แน่นอน
- ☐ ไม่มีข้อใดถูก
- ☒ พิมพ์ว่า value = 10

ล้างสิ่งที่เลือก

โปรแกรมจะแสดงชุดตัวเลขใดออกมา

1 คะแนน

```
#include <stdio.h>

int get_score(void)
{ static int score=0;

  score++;
  return score;
}

int main()
{ int i;

  for (i=1; i < 10; i++)
    printf("%d, ", get_score());
}
```

- ☐ 1, 2, 3, 4, 5, 6, 7, 8, 9,
- ☐ 1, 3, 5, 7, 9, 11, 13, 15, 17,
- ☒ 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
- ☐ 1, 3, 5, 7, 9, 11, 13, 15, 17, 19
- ☐ 2, 3, 4, 5, 6, 7, 8, 9, 10,
- ☐ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
- ☐ 1 2 3 4 5 6 7 8 9 10

หลังจากที่โปรแกรมเริ่มทำงาน หากฟังก์ชัน func ถูกเรียกใช้ บรรทัดที่ 9 จะพิมพ์ข้อความอย่างไรออกมา

1 คะแนน

```
1 #include <stdio.h>
2
3 int value = 10;
4
5 void func()
6 { int value;
7
8   value = value;
9   printf("value = %d\n", value);
10 }
```

- ☐ พิมพ์ว่า value = 0
- ☐ พิมพ์ว่า value = 10
- ☐ ไม่มีข้อใดถูก
- ☒ ค่าของ value ที่ถูกพิมพ์ออกมาเป็นเลขจำนวนเต็มซึ่งมีค่าไม่แน่นอน

ล้างสิ่งที่เลือก

การประกาศตัวแปร global โดยไม่ได้กำหนดค่าเริ่มต้นให้กับตัวแปร ตัวแปรนี้จะมีค่าเริ่มต้นในตัวแปรเป็นอย่างไร 1 คะแนน

- ☐ ตัวแปรจะยังไม่ถูกเก็บในหน่วยความจำ จนกว่าจะมีการกำหนดค่าให้กับตัวแปร
- ☒ 0 เสมอ
- ☐ ค่าไม่แน่นอน ซึ่งขึ้นอยู่กับว่าค่าในหน่วยความจำที่ค้างอยู่จะมีค่าเป็นเท่าใด
- ☐ ไม่มีข้อใดถูก

พวกเราคิดว่าผลลัพธ์ของการรันโปรแกรมนี้เป็นอย่างไร

1 คะแนน

```
#include <stdio.h>
int x;
int main()
{
    for (int x=0; x < 5; x++)
        printf("%d, ", x);
    printf("%d, ", x);
}
```

0, 1, 2, 3, 4, 0,

ตัวแปร local ในฟังก์ชันมีช่วงชีวิต (ที่ถูกเก็บไว้ในหน่วยความจำ) อย่างไร

1 คะแนน

- ☐ มีชีวิตตลอดการทำงานของโปรแกรม แต่หลังจากที่ฟังก์ชันถูกเรียกใช้แล้วอย่างน้อย 1 ครั้ง
- ☐ มีชีวิตตลอดการทำงานของโปรแกรมตั้งแต่โปรแกรมเริ่มทำงานจนกระทั่งสิ้นสุดการทำงาน
- ☐ มีชีวิตเพียงช่วงการทำงานของฟังก์ชัน เมื่อฟังก์ชันทำงานเสร็จ ตัวแปรจะหายไป แต่เมื่อฟังก์ชันถูกเรียกซ้ำอีก ค่าเดิมของตัวแปรจะกลับมาเสมอ
- ☒ มีชีวิตเพียงช่วงการทำงานของฟังก์ชัน เมื่อฟังก์ชันทำงานเสร็จ ตัวแปรจะหายไป
- ☐ ไม่มีข้อใดถูก

หากเราประกาศตัวแปร x ที่เป็นตัวแปร global ไว้ในไฟล์โค้ด src1.c แต่ต้องการเข้าถึงค่าในตัวแปร x นี้ในไฟล์โค้ด src2.c เราควรใช้คีย์เวิร์ดใดในการประกาศตัวแปร x ที่อยู่ในไฟล์ src2.c

1 คะแนน

- ☒ extern
- ☐ auto
- ☐ ไม่จำเป็นต้องใส่คีย์เวิร์ดเพิ่มเติม
- ☐ static
- ☐ register
- ☐ ไม่สามารถเข้าถึงตัวแปรข้ามไฟล์กันได้

ข้อใดกล่าวถูกต้องเกี่ยวกับการใช้ make

1 คะแนน

- ☐ เป้าหมาย (targets) จะเป็นได้เพียงไฟล์ executable ที่ต้องการสร้าง
- ☐ make ใช้ได้กับการคอมไพล์โปรแกรมเท่านั้น
- ☒ make จะตรวจสอบ dependency เพื่อดูว่ากฎใดบ้างจำเป็นต้องถูกเรียกใช้งาน
- ☒ เราสามารถสร้างกฎ (rules) สำหรับการคอมไพล์ไว้ใน Makefile

กลับ

ส่ง

หน้า 2 จาก 2

ห้ามส่งรหัสผ่านใน Google ฟอรัม

แบบฟอร์มนี้ถูกสร้างขึ้นภายใน มหาวิทยาลัยเกษตรศาสตร์ [รายงานการละเมิด](#)

Google ฟอรัม

คุณกำลังแก้ไขคำตอบ การแชร์ URL จะเป็นการอนุญาตให้ผู้อื่นแก้ไขคำตอบของคุณได้ด้วย

เปิดแบบฟอร์มเปล่า

คุณกำลังแก้ไขคำตอบ การแชร์ URL จะเป็นการอนุญาตให้ผู้อื่นแก้ไขคำตอบของคุณได้ด้วย

เปิดแบบฟอร์มเปล่า