

Started on	Friday, 5 June 2020, 10:02 PM
State	Finished
Completed on	Friday, 5 June 2020, 10:11 PM
Time taken	8 mins 31 secs
Marks	6.67/8.00
Grade	8.33 out of 10.00 (83%)

Question **1**
Correct
Mark 1.00 out of 1.00

Cho đoạn chương trình thao tác với biến cấu trúc Point mô tả 1 điểm trong không gian 2 chiều như sau:

```
struct Point a;  
scanf("%lf%lf",&a.x,&a.y); // Nhập từ bàn phím cho tung trục  
printf("(%.3lf, %.3lf)",a.x,a.y); //Hiển thị
```

Hãy viết khai báo cho cấu trúc point(struct Point) để tạo thành 1 chương trình có thể thực thi được.

Chú ý

- Chỉ viết phần khai báo, KHÔNG VIẾT TOÀN BỘ CHƯƠNG TRÌNH

Answer: (penalty regime: 33.3, 66.7, ... %)

```
1 #include <stdio.h>  
2 struct Point{  
3     double x;  
4     double y;  
5 };  
6
```

	Input	Expected	Got	
✓	10 15.5	(10.000, 15.500)	(10.000, 15.500)	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **2**

Correct

Mark 0.67 out of 1.00

Cho trước [hàm](#) `main()` để nhập từ bàn phím tọa độ 1 điểm (Point: gồm 2 thành phần x,y - mỗi thành phần là 1 số thực) trong không gian 2 chiều và hiển thị tọa độ này lên màn hình như sau:

```
int main(){  
  
    struct Point a;  
  
    a = readPoint(); // Gọi hàm nhập 1 điểm từ bàn phím  
  
    printPoint(a); // Hiển thị điểm a lên màn hình  
  
    return 0;  
  
}
```

Viết khai báo cho cấu trúc Point, [hàm](#) nhập điểm và [hàm](#) hiển thị điểm để [hàm](#) `main()` như trên có thể thực thi được.

Nguyên mẫu (Prototype)

- Tên [hàm](#): `readPoint()`
- Tham số: **không**
- Kiểu trả về: **Point**

Thân [hàm](#) (Body)

- Khai báo 1 biến cấu trúc Point, nhập từng trường của biến và trả về kết quả của biến cho [hàm](#)

Nguyên mẫu (Prototype)

- Tên [hàm](#): `printPoint()`
- Tham số: `a` - kiểu Point
- Kiểu trả về: **không**

Thân [hàm](#) (Body)

- Hiển thị từng trường của biến cấu trúc `a`.

Lưu ý:

- Sinh viên chỉ submit phần khai báo kiế cấu trúc Point và các [hàm](#) `readPoint()`, `printPoint()`; không submit [hàm](#) `main()`

For example:

Test	Input	Result
<pre>struct Point a; a = readPoint(); printPoint(a);</pre>	5.57 6.6	(5.570, 6.600)

Answer: (penalty regime: 33.3, 66.7, ... %)

```
1 struct Point{  
2     double x;  
3     double y;  
4 };  
5 struct Point readPoint(){  
6     struct Point a;  
7     scanf("%lf%lf", &a.x, &a.y);  
8     return a;  
9 }  
10 void printPoint(struct Point a){  
11     printf("(%.3lf, %.3lf)", a.x, a.y);  
12 }  
13
```

	Test	Input	Expected	Got	
✓	<pre>struct Point a; a = readPoint(); printPoint(a);</pre>	5.57 6.6	(5.570, 6.600)	(5.570, 6.600)	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.67/1.00**.

Question **3**

Correct

Mark 0.00 out of 1.00

Cho cấu trúc Point có khai báo

```
struct Point{
    double X,Y;
};
```

Viết [hàm](#) tính khoảng cách giữa 2 điểm trong không gian 2 chiều.

Khuôn dạng (Prototype)

- Tên [hàm](#) (function name): **distance**
- Tham số (parameters):
 - A: struct Point
 - B: struct Point
- Kiểu trả về (return type): **double**

Thân [hàm](#) (Body)

- Tính và trả về khoảng cách giữa 2 điểm A(x1, y1) và B(x2,y2) theo công thức:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Chú ý

- Giá trị của các tham số luôn hợp lệ, Không cần kiểm tra.
- Chỉ viết [hàm](#) lowCase, KHÔNG VIẾT TOÀN BỘ CHƯƠNG TRÌNH
- Xem thêm chi tiết trong phần **For example**.

Chú ý

- Chỉ viết phần khai báo, KHÔNG VIẾT TOÀN BỘ CHƯƠNG TRÌNH

For example:

Test	Result
struct Point A={1,2}; struct Point B={3,4}; printf("%.4f",distance(A,B));	2.8284
struct Point A={1,10}; struct Point B={3,4}; printf("%.4f",distance(A,B));	6.3246

Answer: (penalty regime: 33.3, 66.7, ... %)

```
1 | #include <stdio.h>
2 | #include <math.h>
3 | // struct Point{
4 | //     double X,Y;
5 | // };
6 | double distance(struct Point a, struct Point b){
7 |     return sqrt(pow(a.X - b.X, 2) + pow(a.Y - b.Y, 2));
8 | }
```

Test	Expected	Got
------	----------	-----

	Test	Expected	Got	
✓	struct Point A={1,2}; struct Point B={3,4}; printf("%.4f",distance(A,B));	2.8284	2.8284	✓
✓	struct Point A={1,10}; struct Point B={3,4}; \t printf("%.4f",distance(A,B));	6.3246	6.3246	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.00/1.00**.

Question **4**

Correct

Mark 1.00 out of 1.00

Cho cấu trúc Point có khai báo

```
struct Point{
    double X,Y;
};
```

Viết [hàm](#) kiểm tra 2 điểm A, B có giống nhau không?

Với các số thực, không nhất thiết kiểm tra các tọa độ X,Y của các điểm A,B chính xác là giống nhau; thay vào đó ta kiểm tra khoảng cách của chúng có nhỏ hơn một sai số Epsilon rất nhỏ cho trước hay không (chẳng hạn Epsilon=0.0000001)

Khuôn dạng (Prototype)

- Tên [hàm](#) (function name): **equal**
- Tham số (parameters):
 - A: struct Point
 - B: struct Point
 - Epsilon: double
- Kiểu trả về (return type): **int**

Thân [hàm](#) (Body)

- Nếu 2 điểm giống nhau, kết quả là 1, ngược lại kết quả là 0

Chú ý

- Giá trị của các tham số luôn hợp lệ, Không cần kiểm tra.
- Chỉ viết [hàm](#), KHÔNG VIẾT TOÀN BỘ CHƯƠNG TRÌNH
- Xem thêm chi tiết trong phần **For example**.

Chú ý

- Chỉ viết phần khai báo, KHÔNG VIẾT TOÀN BỘ CHƯƠNG TRÌNH

For example:

Test	Result
struct Point A={1,2}; struct Point B={1,3}; if (equal(A,B, 0.000001)) printf("A = B"); else printf("A != B");	A != B
struct Point A={1,2}; struct Point B={1,2.00000001}; if (equal(A,B, 0.000001)) printf("A = B"); else printf("A != B");	A = B

Answer: (penalty regime: 33.3, 66.7, ... %)

```
1  #include <stdio.h>
2  #include <math.h>
3  double distance(struct Point a, struct Point b){
4      return sqrt(pow(a.X - b.X, 2) + pow(a.Y - b.Y, 2));
5  }
6  int equal(struct Point a, struct Point b, double ep){
7      return (distance(a,b) <= ep);
8  }
```

	Test	Expected	Got	
✓	<pre>struct Point A={1,2}; struct Point B={1,3}; \t if (equal(A,B, 0.000001)) \tprintf("A = B"); else printf("A != B");</pre>	A != B	A != B	✓
✓	<pre>struct Point A={1,2}; struct Point B={1,2.00000001}; \t if (equal(A,B, 0.000001)) \tprintf("A = B"); else printf("A != B");</pre>	A = B	A = B	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **5**

Correct

Mark 1.00 out of 1.00

Cho cấu trúc Point có khai báo

```
struct Point{
    double X,Y;
};
```

Viết [hàm](#) kiểm tra tọa độ của 3 điểm A,B,C đầu vào có tạo thành 1 tam giác hay không?

Điều kiện tạo thành tam giác là độ dài tổng 2 cạnh bất kỳ phải lớn hơn độ dài cạnh còn lại

Khuôn dạng (Prototype)

- Tên [hàm](#) (function name): **isTriangle**
- Tham số (parameters):
 - A: struct Point
 - B: struct Point
 - C: struct Point
- Kiểu trả về (return type): **int**

Thân [hàm](#) (Body)

- Nếu 3 điểm có thể tạo thành tam giác, kết quả là 1, ngược lại kết quả là 0

Chú ý

- Giá trị của các tham số luôn hợp lệ, Không cần kiểm tra.
- Chỉ viết [hàm](#) lowCase, KHÔNG VIẾT TOÀN BỘ CHƯƠNG TRÌNH
- Xem thêm chi tiết trong phần **For example**.

Chú ý

- Chỉ viết phần khai báo, KHÔNG VIẾT TOÀN BỘ CHƯƠNG TRÌNH

For example:

Test	Result
<pre>struct Point A={1,2}; struct Point B={1,3}; struct Point C={2,3}; if (isTriangle(A,B,C)) printf("YES"); else printf("NO");</pre>	YES

Answer: (penalty regime: 33.3, 66.7, ... %)

```
1  #include <stdio.h>
2  #include <math.h>
3  double distance(struct Point a, struct Point b){
4      return sqrt(pow(a.X - b.X, 2) + pow(a.Y - b.Y, 2));
5  }
6  int isTriangle(struct Point a, struct Point b, struct Point c){
7      return ((distance(a,b) + distance(a,c) > distance(b,c)) && (distance(b,a) + distance(b,c) > distance(a,b)) && (distance(a,b) + distance(b,c) > distance(a,c)));
8  }
```

	Test	Expected	Got	
--	------	----------	-----	--

	Test	Expected	Got	
✓	<pre>struct Point A={1,2}; struct Point B={1,3}; struct Point C={2,3}; \t if (isTriangle(A,B,C)) \tprintf("YES"); else printf("NO");</pre>	YES	YES	✓
✓	<pre>struct Point A={1,2}; struct Point B={1,3}; struct Point C={1,10}; \t \tif (isTriangle(A,B,C)) \t\tprintf("YES"); \telse printf("NO");</pre>	NO	NO	✓

Passed all tests! ✓

Correct

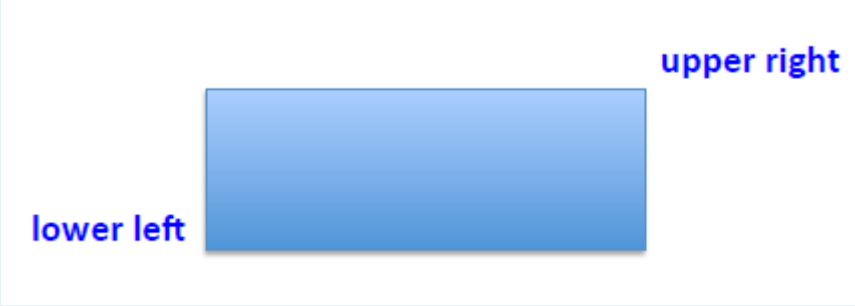
Marks for this submission: 1.00/1.00.

Question **6**

Correct

Mark 1.00 out of 1.00

Định nghĩa kiểu hình chữ nhật Rect ở đó các cạnh song song với các trục tọa độ. Một hình chữ nhật có thể được biểu diễn bằng tọa độ điểm trên phải và dưới trái của kiểu Point



Gợi ý: Định nghĩa kiểu Rect với 2 thành viên là điểm trên phải và điểm dưới trái

Chú ý

- Chỉ viết phần khai báo, KHÔNG VIẾT TOÀN BỘ CHƯƠNG TRÌNH

For example:

Test	Result
struct Rect a={{5,7},{1,10}};	YES

Answer: (penalty regime: 33.3, 66.7, ... %)

```
1 #include <stdio.h>
2 // struct Point{
3 //   double x,y;
4 // };
5 struct Rect{
6     struct Point upperRight,lowerLeft;
7 };
```

	Test	Expected	Got	
✓	struct Rect a={{5,7},{1,10}};	YES	YES	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **7**

Correct

Mark 1.00 out of 1.00

Khai báo kiểu hình chữ nhật Rect và viết [hàm](#) tính diện tích hình chữ nhật.

Khuôn dạng (Prototype)

- Tên [hàm](#) (function name): **area**
- Tham số (parameters):
 - A: struct Rect
- Kiểu trả về (return type): **double**

Thân [hàm](#) (Body)

- Tính và trả về diện tích của hình chữ nhật

Chú ý

- Giá trị của các tham số luôn hợp lệ, Không cần kiểm tra.
- Xem thêm chi tiết trong phần **For example**.

Chú ý

- Chỉ viết khai báo kiểu và định nghĩa [hàm](#), KHÔNG VIẾT TOÀN BỘ CHƯƠNG TRÌNH

For example:

Test	Result
struct Rect r = {{10,2},{2,8}}; printf("%.5lf",area(r));	48.00000
struct Rect r = {{10,2.5},{2,8.3}}; printf("%.5lf",area(r));	46.40000

Answer: (penalty regime: 33.3, 66.7, ... %)

```
1  #include <stdio.h>
2  #include <math.h>
3  struct Point{
4      double x,y;
5  };
6  struct Rect{
7      struct Point lowerLeft, upperRight;
8  };
9  double distance(struct Point a, struct Point b){
10     return sqrt(pow(a.x - b.x, 2) + pow(a.y - b.y, 2));
11 }
12 double area(struct Rect a){
13     struct Point temp = {a.lowerLeft.x, a.upperRight.y};
14     return distance(a.lowerLeft, temp) * distance(temp, a.upperRight);
15 }
```

	Test	Expected	Got	
✓	struct Rect r = {{10,2},{2,8}}; printf("%.5lf",area(r));	48.00000	48.00000	✓
✓	struct Rect r = {{10,2.5},{2,8.3}}; printf("%.5lf",area(r));\t	46.40000	46.40000	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **8**

Correct

Mark 1.00 out of 1.00

Viết [hàm](#) kiểm tra một điểm có thuộc 1 hình chữ nhật cho trước hay không?

Khuôn dạng (Prototype)

- Tên [hàm](#) (function name): **isIn**
- Tham số (parameters):
 - M: struct Point
 - r: struct Rect
- Kiểu trả về (return type): **int**

Thân [hàm](#) (Body)

- Nếu M nằm trong hình chữ nhật r, kết quả của [hàm](#) là 1, ngược lại kết quả là 0.

Chú ý

- Giá trị của các tham số luôn hợp lệ, Không cần kiểm tra.
- Xem thêm chi tiết trong phần **For example**.

Gợi ý

- Với giả thiết là các cạnh của hình chữ nhật song song với các trục tọa độ, giả sử 2 điểm tọa độ mô tả hình chữ nhật là A(top right) và B (bottom left)
 - Hình chiếu của A, B, M xuống Ox có tọa độ lần lượt là $Ax(x_A, 0)$, $Bx(x_B, 0)$, $Mx(x_M, 0)$.
 - Kiểm tra $Mx \in Ax Bx$ hay không. Nếu $0 \leq \frac{x_M - x_A}{x_B - x_A} \leq 1$ thì $Mx \in Ax Bx$, ngược lại $Mx \notin Ax Bx$
 - Hình chiếu của A, B, M xuống Oy có tọa độ lần lượt là $Ay(0, y_A)$, $By(0, y_B)$, $My(0, y_M)$.
 - Kiểm tra $My \in Ay By$ hay không. Nếu $0 \leq \frac{y_M - y_A}{y_B - y_A} \leq 1$ thì $My \in Ay By$, ngược lại $My \notin Ay By$
- M thuộc hình chữ nhật r khi điều kiện kiểm tra ở 2 và điều kiện kiểm tra ở 4 đều đúng.

For example:

Test	Result
<pre>struct Rect r = {{10,2},{2,8}}; struct Point M={10.01,6}; if (isIn(M,r)) printf("YES"); else printf("NO");</pre>	NO
<pre>struct Rect r = {{10,2},{2,8}}; struct Point M={8,6}; if (isIn(M,r)) printf("YES"); else printf("NO");</pre>	YES

Answer: (penalty regime: 33.3, 66.7, ... %)

```
1  #include <stdio.h>
2  #include <math.h>
3  struct Point{
4      double x,y;
5  };
6  struct Rect{
7      struct Point A, B;
8  };
9  double distance(struct Point a, struct Point b){
10     return sqrt(pow(a.x - b.x, 2) + pow(a.y - b.y, 2));
11 }
12 int isIn(struct Point m, struct Rect r){
13     struct Point  bx = {r.B.x, 0};
14     struct Point  ax = {r.A.x, 0};
15     struct Point  mx = {m.x, 0};
16     struct Point  by = {0, r.B.y};
17     struct Point  ay = {0, r.A.y};
18     struct Point  my = {0, m.y};
19     return (distance(ax, mx) + distance(bx, mx) == distance(ax, bx)) && (distance(ay, my) -
20 }
```

	Test	Expected	Got	
✓	<pre>struct Rect r = {{10,2},{2,8}}; struct Point M={10.01,6}; \t if (isIn(M,r)) printf("YES"); else printf("NO");</pre>	NO	NO	✓
✓	<pre>struct Rect r = {{10,2},{2,8}}; struct Point M={8,6}; \t if (isIn(M,r)) \tprintf("YES"); else printf("NO");</pre>	YES	YES	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[◀ Khai báo cấu trúc](#)

Jump to...

⬆

[Phân số ▶](#)