

$$0+1+1=2 \quad x=2$$

$$2+2+3=8 \quad x=8$$

①

王俊崑 22302016002

第一部分:

(1) 存储器

(2) 循环

(3) 24

(4) 4

(5)

(6) 1 或 true

(7) $y \% 2 \neq 0$

(8) 1

(9) $i - (i/3) - 1$

(10) -6

$i - (i/3) - 1$

$$m = 5 \cdot 2 = 10 \quad 10 - 10 = -8 \quad 2 + (-8) = -6$$

第二部分

(1) A

(2) C

(3) C

(4) A

(5) B

(6) B

(7) D

(8) D

(9) B

(10) D

第三部分

(1) print the default value

(2) $a = 9$

$b = 5$

(3) 第九行,

第一行

printf("三角形为直角三角形");

used sqrt without dedaring math library

修正: printf("三角形为直角三角形\n"); 修正: #include <math.h>

第十行

用 %d 是错误的, 因为 s 和 p 变量

是 float

修正: printf("三角形面积: %.2f, 三角形周长: %.2f\n", s, 2 * p);

第四行: extra → 也可以把 if 改成 while, 把 return 去掉 while () {

不可以输入少于三个数, 但可以

超过三个 → 修正: if (scanf("%f %f %f", a, b, c) != 3)

printf("输入错误\n");
}



②

$a=10$
 $b=a++ \rightarrow b=10, a=11$
 $c=++a \rightarrow c=12, a=12$
 $b=10 * a++ \rightarrow d=10 * 12 = 120, a=13$

(4) 13 10 12 120

(5)

1			
2	4		
3	9	27	
4	16	64	256
-			



- (6) (1) $i < 4$ (4) $(i \neq j) \Delta \Delta (j \neq k)$
 (2) $j < 4$ (5) $(i+j+k) \% 5 == 0$
 (3) $k < 4$

第四部分

(1) #include <stdio.h>

int main(void) {

float score[3][4] = {

{65, 57, 70, 60},

{90, 99, 100, 98},

{50, 87, 90, 87}

};

int i;

for (i=0; i<3; i++) {

int flag = 0;

for (int j=0; j<4; j++) {

if (score[i][j] < 60)

flag = 1;

}

if (flag == 1) {

printf("No. %d fails, his scores are: \n", i+1);

for (int j=0; j<4; j++) printf("%f ", score[i][j]);

printf("\n");

}

}
return 0; }



④

```
(2) int randInt (int min, int max) {  
    return rand() / (max - min + 1) + min;
```

```
}
```

```
void display() {
```

```
    for (int i = 0; i < SIZE; ++i) {
```

```
        for (int j = 0; j < SIZE; ++j) {
```

```
            printf("| "); // 后面一个空格
```

```
            if (board[i][j] == 1) printf("*");
```

```
            else printf(" "); // 一个空格
```

```
        }
```

```
        printf(" |"); // 前面一个空格
```

```
        printf("\n");
```

```
    }
```

```
    return;
```

```
}
```

```
int judgeKnock (int x, int y) {
```

```
    if (x > 9 || y > 9) return 1;
```

```
    else if (x < 0 || y < 0) return 1;
```

```
    return 0;
```

```
}
```




```
int main (void) {
```

```
    srand((uint) time(NULL));
```

```
    int direction;          int vector [4][2] = { {1,0}, {0,1}, {-1,0}, {0,-1} };
```

```
    int pX, pY;
```

```
    pX = randInt(0,9);
```

```
    pY = randInt(0,9);
```

```
    direction = randInt(0,3);
```

```
    int maxSteps;
```

```
    printf("Please enter max steps; \n");
```

```
    scanf("%d", &maxSteps);
```

```
    for(int i=0; i < maxSteps; i++) {
```

```
        if (board[pY][pX] == 1) {
```

```
            board[pY][pX] = 0;
```

```
            if (direction - 1 ≤ 0) {
```

```
                direction = 4;
```

```
            } else direction--;
```

```
        } else if (board[pY][pX] == 0) {
```

```
            board[pY][pX] = 1;
```

```
            if (direction + 1 ≥ 4) direction = 0;
```

```
            else direction++;
```

```
        }
```

```
        pY = pY + vector[direction][0]; pX = pX + dir vector[direction][1];
```

```
        if (judgeKnock(pX, pY) == 1) break;
```

```
    }
```

```
    display();
```

```
    return 1; }
```



Scrap paper 王俊崑

第三部分 (2)

$b=1, a=1$	$b=2, a=3$	$b=3, a=5$	$b=4, a=7$	$b=5, a=9$
$\begin{array}{r} 0 \\ 2 \overline{) 1} \\ \underline{0} \\ 1 \end{array}$	$\begin{array}{r} 1 \\ 2 \overline{) 3} \\ \underline{2} \\ 1 \end{array}$	$\begin{array}{r} 2 \\ 2 \overline{) 5} \\ \underline{4} \\ 1 \end{array}$	$\begin{array}{r} 3 \\ 2 \overline{) 7} \\ \underline{6} \\ 1 \end{array}$	break
$1 \cdot 2 = 1$	$3 \cdot 2 = 1$	$5 \cdot 2 = 1$	$7 \cdot 2 = 1$	
$a+=5 \Rightarrow a=6$	$a+=5 \Rightarrow 8$	$a+=5 \Rightarrow a=10$	$a+=5 \Rightarrow 12$	
$a-=3 \Rightarrow a=3$	$a-=3 \Rightarrow 5$	$a-=3 \Rightarrow 7$	$a-=3 \Rightarrow 9$	

(5) $i=1$	$i=2$	$i=3$	$i=4$	$\begin{array}{r} 16 \\ \times 4 \\ \hline 64 \end{array}$	$\begin{array}{r} 64 \\ \times 4 \\ \hline 256 \end{array}$
$\rightarrow 1$	$1 \cdot 2 = 2$	$1 \cdot 3 = 3$	$1 \cdot 4 = 4$		
	$2 \cdot 2 = 4$	$3 \cdot 3 = 9$	$4 \cdot 4 = 16$		
	$\rightarrow 2 \quad 4$	$9 \cdot 3 = 27$	$16 \cdot 4 = 64$		
		$\rightarrow 3 \quad 9 \quad 27$	$64 \cdot 4 = 256$		
			$\rightarrow 4 \quad 16 \quad 64 \quad 256$		

