

《程序设计 (I)》期末试题参考答案 (A 卷)

Section A: Multiple Choices (20 points)

1-5: (A) (A) (D) (D) (A)

6-10: (A) (C) (B) (C) (C)

Section B: Short Answer (40 points)

1.

Ans:

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

(b) $x = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$ (2points)

(c) $x = \frac{\text{fabs}(c)}{\sqrt{a^2 + b^2}}$ (2points)

2.

```
double pi(int n) {
    int i;
    double sum = 0;

    for (i = 1; i <= n; i++) {
        sum += 1.0 / (i * i); (3 points)
    }
    return sqrt(6 * sum); (3 points)
}
```

3.

(a) (2 points)

```
struct circle {
    struct point p;
    double r;
};
```

(b) (2 points)

```
double getArea(struct circle c) {
    return 3.14159 * c.r * c.r;
}
```

(c) (2 points)

```
double getCircumference(struct circle c) {  
    return 2 * 3.14159 * c.r;  
}
```

(d) (4 points)

```
int isInside(struct point p, struct circle c) {  
    if ((p.x - c.p.x) * (p.x - c.p.x) + (p.y - c.p.y) * (p.y - c.p.y) < c.r * c.r)  
        return 1;  
    else  
        return 0;  
}
```

4.

```
void dec2bin(int n) {  
    int b[64], i;  
  
    for (i = 0; n > 0; i++) {  
        b[i] = n % 2; (3 points)  
        n /= 2; (2 points)  
    }  
    while (i-- > 0) printf("%d", b[i]); (3 points)  
}
```

5.

(a) (5 points)

```
double getMax(double a[], int length) {  
    int i;  
    double max = a[0];  
  
    for (i = 1; i < length; i++) {  
        if (a[i] > max) max = a[i];  
    }  
    return max;  
}
```

(b) (5 points)

```
int reverse(int a) { /* ignore the case of negative integers */  
    int b = 0;  
  
    while (a > 0){  
        b = b * 10 + a % 10;  
        a /= 10;  
    }  
}
```

```
    return b;
}
```

Or:

```
int reverse(int a) { /* ignore the case of negative integers */
    int b = 0;

    do {
        b += a % 10;
        b *= 10;
    } while ((a /= 10) >= 1);
    return b / 10;
}
```

Or other correct answer.

```
int reverse(int a) { /* ignore the case of negative integers */
    int b = 0;

    do {
        b += a % 10;
        b *= 10;
    } while ((a /= 10) >= 1);
    return b / 10;
}
```

Section C: Program Output Analysis (20 points)

1.

a = 8, b = 4.	(2 points)
x = 18, y = 4.	(2 points)
a = 20, b = 20.	(2 points)
x = 21, y = 441.	(2 points)

2.

sum[0] = 0	(1 point)
sum[1] = 1	(1 point)
sum[2] = 3	(1 point)
sum[3] = 6	(1 point)
sum[4] = 10	(1 point)
sum[5] = 15	(1 point)

3.

310	(3 points)
321	(3 points)

Section D: Program Error Correction (20 points)

1.

(a) (3 points) Error: constant should be **const** (1 point). The string should have double quotation marks (1 point). If it is constant, the value cannot be changed.

Correction:

```
char *str[] = {"Hi", "Kay"};
str[1] = "Joe"; /* 1 point for deleting "*" */
```

(b) (3 points) Error: The array initialization should have braces (1 point). `b[i, i]` should be `b[i][i]` (1 point). The indices of the last element are 5 and 5, or the size of the array should be 7 (1 point).

Correction:

```
int b[6][6] = {0}, i;
for (i = 0; i <= 5; i++) b[i][i] = i * i;
```

(c) (4 points) Error: The function should have a return data type and value (2 points). The explicit type casting on `x` is unnecessary (but not an error). The two function parameters should be declared separately (2 points).

Correction:

```
double add(double x, double y) {
    return x + y;
}
```

2.

```
#1:  #include <stdio.h>
#2:  int main() {
#3:      double base, height, area;
#4:
#6:      printf("Enter base, height: ");
#7:      scanf("%lf, %lf", &base, &height);           // 3 points
#5:      area = (1 / 2.0) * base * height;           // 3 points
#8:      printf("Area = %lf\n", area);                // 2 points
      // or printf("Area = %f\n", area);
#9:      return 0;
#10: }
```

2 points for knowing rectifying the orders of statement 6,7 and 5.