中山大学软件学院 2009 级软件工程专业 (2009 秋季学期)

《程序设计（I)》期末试题（A卷)

（考试形式：闭卷　考试时间：2 小时）

警示

《中山大学授予学士学位工作细则》第六条

考试作弊不授予学士学位

方向：_____　姓名：_____　学号：_____

Section A: Multiple Choice (20 points)

For each of the following questions, choose only ONE of the provided multiple-choice: A, B, C, D corresponding to the best answer for them.

1. Which is an example of unary operators?

(A) ++
(B) <=
(C) =
(D) &&

2. How many times will the following program fragment print *hello*?

```
for (i = 2; i < 1000; i *= i)  printf("hello");
```

(A) 4
(B) 5
(C) 6
(D) None of the above

3. A recursive function is a function that

(A) returns itself
(B) takes a function as an argument
(C) is inside of another function
(D) calls itself

4. Which is true?

(A) An array can contain data items of different data types.
(B) An array size can be changed after declaration.
(C) The subscript for the last element of an array is the array size.
(D) None of the above.

5. What is NOT a benefit of functions?

(A) Make a program faster
(B) Software reusability
(C) Avoid code repetition
(D) Divide and conquer

6. Assume `hello` is a character array. Which of the following operations does NOT produce a string?

(A)
```
char hello[] = {'h', 'e', 'l', 'l', 'o'};
```
(B)
```
char hello[] = {'h', 'e', 'l', 'l', 'o', '\0'};
```
(C)
```
char hello[] = " ";
```
(D)
```
char hello[] = "hello";
```

7. The binary search algorithm

(A) is better suited to small arrays
(B) is better suited to unsorted arrays
(C) can only be used on a sorted array
(D) is slower than a linear search

8. If `bPtr` is assigned `b` (the name of an array), then array element `b[6]` can alternatively be referenced as:

(A)
```
b[bPtr + 6]
```
(B)
```
*(bPtr + 6)
```
(C)
```
*b[bPtr + 6]
```
(D)
```
bPtr + 6
```

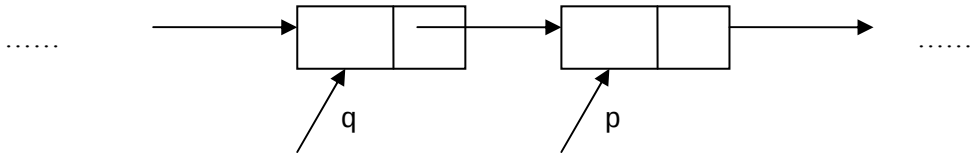
9. Which statement will NOT cause compilation error?

(A)
```
int a[][] = {1, 2, 3};
```
(B)
```
int a[2][] = {1, 2, 3};
```
(C)
```
int a[][2] = {1, 2, 3};
```
(D) None of the above

10. Suppose we have the following node definitions and declaration.

```
struct node {
   int a;
   struct node *link;
} *head, *p, *q;
```

`p` and `q` respectively point to two adjacent nodes , shown as following:



Which of the following statements are NOT correct statements to delete node p?

(A) `q->link = p->link; free(p);`

(B) `p = p->link; free(q->link); q->link = p;`

(C) `(*p).link = (*q).link; free(p);`

(D) `p = (*p).link; free((*q).link); (*q).link = p;`

Section B: Short Answer (40 points)

Briefly answer the questions according the requirements.

1. (6 points) Represent the following three equations in C programming language.

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

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

(c) (2 points) $x = \dfrac{|c|}{\sqrt{a^2 + b^2}}$. Hint: `fabs()` is a function for the absolute value.

2. (6 points) The value of π can be calculated by the following formula:

$$\pi = \sqrt{6(\frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + ... + \frac{1}{n^2})}$$

The following function **`double pi(int n)`** is to use this formula to calculate the π and return the value. Please write ONE C statement in each blank to complete the function definition:

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

   for (i = 1; i <= n; i++) {

      _____________

   }
```

```
    return _________
}
```

3. (10 points) A structure for a point is defined as following specifications:

```
struct point {
    double x;
    double y;
};
```

(a) (2 points) Use the above `point` structure to declare a `circle` structure for a circle which has two members: the center point `p` and the radius `r`.
(b) (2 points) Write the function `getArea()` that passes a circle as the parameter and returns the area of the circle.
(c) (2 points) Write the function `getCircumference()` that passes a circle as the parameter and returns the circumference of the circle.
(d) (4 points) Write the function `isInside()` that passes a point and a circle as the parameters and decides if the point is inside the circle or not. If the point is inside the circle, returns 1 and otherwise return 0.

4. (8 points) The following function is to write a function **`void dec2bin(int n)`** that print the binary format of an non-negative integer `n`. For example, 123 = 1111011_2.
Please write ONE C statement in each blank to complete the function definition:

```
void dec2bin(int n) {
    int b[64], i;

    for (i = 0; n > 0; i++) {
        __________
        __________
    }
    while ( _____ )  printf("%d", b[i]);
}
```

5. (a) (5 points) Write the function `getMax()` that pass a **`double`** array and its size and returns the maximum value of the array elements.
(b) (5 points) Write the function `reverse()` that pass an integer and return a value that reverses it. For example, if 1234 is passed into the function, 4321 will be returned.

Section C: Program Output Analysis (20 points)

Write the result after executing the following programs or program fragments.

1. (8 points)

```
#include <stdio.h>

int func(int a, int b){
   a *= 2;
   printf("a = %d, b = %d.\n", a, b);
   return (a - 2) * --b;
}

int sub(int *a, int *b) {
   *a += 2;
   printf("a = %d, b = %d.\n", *a, *b);
   return ++*a * *b--;
}

int main() {
   int x = 3, y = 4;

   x = func(y, y);
   printf("x = %d, y = %d.\n", x, y);
   y = sub(&x, &x);
   printf("x = %d, y = %d.\n", x, y);
   return 0;
}
```

2. (6 points)

```
int i = 0, sum =0;

while (i <= 5) {
   sum += i;
   printf("sum[%d] = %d\n", i, sum);
   i++;
}
```

3. (6 points)

```
#include <stdio.h>

int vtest(int n) {
   static int x = 100;
   int y = 200;

   return n + (x++) + (y++);
}
```

```
int main() {
   printf("%d\n", vtest(10));
   printf("%d\n", vtest(20));
   return 0;
}
```

Section D: Program Error Correction (20 points)

1. (10 points) Identify and correct the errors in each of the following statements or program fragments:

(a) (3 points)

```
constant char *str[] = {'Hi', 'Kay'};
*str[1] = 'Joe';
```

(b) (3 points)

```
int b[6, 6] = 0, i;
for (i = 1; i <= 6; i++)  b[i, i] = i * i;
```

(c) (4 points)

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

2. (10 points) Examine this program:

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

This program is supposed to get the base and height of a triangle, and calculate its area. But it is severely broken. Please fix it.