

试题一、单选题（每小题 3 分，共 30 分）

- | | | | | |
|------------|------------|------------|------------|-------------|
| 1 <u>C</u> | 2 <u>A</u> | 3 <u>D</u> | 4 <u>C</u> | 5 <u>C</u> |
| 6 <u>B</u> | 7 <u>D</u> | 8 <u>D</u> | 9 <u>B</u> | 10 <u>A</u> |

试题二、改错题（共 18 分）

- 1 将 StudentListNode * next; 改为:
struct StudentListNode * next;

将 printf("%d", num % base);
 change(num / base, base);
的顺序交换, 改为:
 change(num / base, base);
 printf("%d", num % base);

-
- 3 删去#define DEBUG 0,
或 #ifdef DEBUG 改为 #if DEBUG

-
- 4 将 return Find(x+m+1, n, v); 改为:
return Find(x+m+1, n-m-1, v);

-
- 5 删除 free(a.name)
或将
 free(a.name);
 printf("%s %d\n", b.name, b.score);
的顺序交换, 改为:
 printf("%s %d\n", b.name, b.score);
 free(a.name);

-
- 6 把 pfore -> next = p; 改为:
If (pfore != NULL)
 pfore -> next = p;

试题三、问答题（共 10 分）

- 相同点: 生命周期都是全程的, (初始化缺省值为 0, 变量只能被初始化一次);
- 1 不同点: 静态局部变量的作用域在函数内部, 静态全局变量的作用域是模块内部

自定义开始，全局变量的作用域还可以扩展到其它模块，只要它们采用 **extern** 声明。

从表上数据的“趋势”看:

- 2 $f_A(2n) \approx 4f_A(n)$ 由此可知，算法 A 的时间复杂度 $T(A) \approx O(n^2)$
 $f_B(2n) \approx 2f_B(n)$ 由此可知，算法 B 的时间复杂度 $T(B) \approx O(n)$
-

试题四、填空题（每小题 2 分，共 24 分）

- | | |
|--|----------------------------|
| 1 $f(n-2)+f(n-1)$ | 2 $a[0] = f(n)$ |
| _____ | _____ |
| 3 $g(a, n)$ | 4 $i=(i+1)\%n$ |
| _____ | _____ |
| 5 find_next(start,monkey,n) | 6 monkey[1][i] = 0 |
| _____ | _____ |
| 7 $(j \geq m) \ \&\& \ (list[j] > key)$ | 8 $i < j$ |
| _____ | _____ |
| 9 swap(&list[m],&list[j]) | 10 map[r][c] == '#' |
| _____ | _____ |
| 11 begin.row = r+d[k][0] | 12 map[r][c] = '=' |
| _____ | _____ |

试题五、算法设计（共 18 分）

1. (1) void input(int **pa, int row, int col)

```
{
    int i,j,k;
    for(i=0; i<row; i++)
        for(j=0; j<col; j++)
            scanf("%d%d", pa[i]+j);
}
```

(2) void mmult(int **pc, int **pa, int row,int col, int **pb)

```
{
    int i,j,k;
    for (i = 0; i < row; ++i){
```

```

        for (j = 0; j < col; ++j){
            pc[i][j] = 0;
            for (k = 0; k < col; ++k){
                pc[i][j] += pa[i][k] * pb[k][j];
            }
        }
    }
}

```

(3) 该算法的时间复杂度为 $O(M*N*N)$,

2. (1)

```

(2) void eraseBlob(int row, int col){
    if (row >= 0 && row < HEIGHT && col >= 0 && col < WIDTH) {
        if (map[row][col] == BLACK) {
            map[row][col] = WHITE;
            eraseBlob(row-1, col); //上
            eraseBlob(row+1, col); //下
            eraseBlob(row, col-1); //左
            eraseBlob(row, col+1); //右
            eraseBlob(row-1, col-1); //左上
            eraseBlob(row+1, col-1); //左下
            eraseBlob(row-1, col+1); //右上
            eraseBlob(row+1, col+1); //右下
        }
    }
}

```