

```

//===== 【第 1 题】 =====

#import <Foundation/Foundation.h>

// 方法声明

NSString * stringCalc(NSString *, NSString *);

int main(int argc, char const *argv[])
{
    NSString *s1 = @"342";
    NSString *s2 = @"17";
    NSString *result = stringCalc(s1, s2);
    NSLog(@"%@ - %@ = %@", s1, s2, result);
    return 0;
}

// 方法实现

NSString * stringCalc(NSString *s1, NSString *s2)
{
    // NSString 的 intValue 方法可以把数字字符串转换成 int 类型的整数
    int a = [s1 intValue] - [s2 intValue];

    // 可以利用 NSString 的 stringWithFormat: 方法创建一个数字字符串
    NSString *s3 = [NSString stringWithFormat:@"%d", a];
    return s3;
}

//===== 【第 2 题】 =====

#import <Foundation/Foundation.h>

// 方法声明

void stringOperate(char *);

int main(int argc, char const *argv[])
{
    char string[] = "www-it-heima-com";
    stringOperate(string);
}

```

```

    printf("%s\n", string);

    return 0;
}

// 方法实现
void stringOperate(char *string)
{
    // 获取字符串长度
    int len = (int)strlen(string);

    // 遍历字符串, 把 www 之后的- 和 com 之前的- 替换成 .
    for (int i = 0; i < len; i++) {
        if (string[i] == '-') {
            if (i == 3 || i == len - 4) {
                string[i] = '.';
            }
        }
    }
    // 替换后的字符串为 www.it-heima.com

    // 再次遍历数组, 找到-, 之后的字符依次往前移动, 覆盖前面的字符
    for (int i = 0; i < len; i++) {
        if (string[i] == '-') {
            int j;
            for (j = i; j < len - 1; j++) {
                string[j] = string[j + 1];
            }
            // 循环结束后 字符串变成 www.it-heima.comm 因为最后一个字符好保留着
            // 把最后一个重复的字符替换成 \0 不输出
            string[j] = '\0';
        }
    }
}

//===== 【第 3 题】 =====

#import <Foundation/Foundation.h>

```

```

// 方法声明
void copyFile(NSString *, NSString *);

int main(int argc, char const *argv[])
{
    // 文件路径用字符串保存
    NSString *f1 = @"/Users/hp/Desktop/sss/file1.m";
    NSString *f2 = @"/Users/hp/Desktop/sss/file2.m";
    copyFile(f1, f2);
    return 0;
}

// 方法实现
void copyFile(NSString *fileToBeCopied, NSString *fileCopied)
{
    // 利用文件内容初始化字符串, fileToBeCopied 是文件路径
    NSString *s = [NSString stringWithContentsOfFile:fileToBeCopied
encoding:NSUTF8StringEncoding error:nil];

    // 把字符串写入文件, fileCopied 为复制好的文件路径
    [s writeToFile:fileCopied atomically:YES encoding:NSUTF8StringEncoding error:nil];
}


//===== 【第 4 题】 =====
#import <Foundation/Foundation.h>

// 方法声明
NSString * dateConver(NSString *date);

int main(int argc, char const *argv[])
{
    NSString *date = @"2013年05月05日";
    NSLog(@"转换前: %@", date);
}

```

```

        NSLog(@"转换后: %@", dateConver(date));

        return 0;
    }

// 方法实现

NSString * dateConver(NSString *date)
{
    // [date rangeOfString:@"年"] 返回查询字符的范围

    // 将查询范围的字符替换成指定字符串

    date = [date stringByReplacingCharactersInRange:[date rangeOfString:@"年"]
withString:@"-"];

    date = [date stringByReplacingCharactersInRange:[date rangeOfString:@"月"]
withString:@"-"];

    date = [date stringByReplacingCharactersInRange:[date rangeOfString:@"日"]
withString:@""];

    return date;
}

//===== 【第 5 题】 =====

#import <Foundation/Foundation.h>

void guessNum();

int main(int argc, char const *argv[])
{
    guessNum();

    return 0;
}

void guessNum()
{
    int num;

    // 以当前时间为随机数种子，每次的随机数都不同

    srand((unsigned)time(NULL));

```

```
// 生产一个[0 9]范围内的随机数

int randNum = rand()%10;

// NSLog(@"生成的随机数是: %d", randNum);

do {

    NSLog(@"请输入一个0-9的整数: ");

    scanf("%d", &num);

} while (num != randNum);

NSLog(@"恭喜你猜对了! ");

}

// 第 6 题

// NSArray 只能保存 OC 对象, 不能保存非 OC 对象, nil 等
```