

# TEST 01

说明：本次测验时间为最长150分钟，满分100分，在答题过程中可以适当使用信息检索能力寻找相关资料，使用非自己所写的代码时，请注意引用的代码量不得超过整体代码量的50%，并注意在引用的代码段前后用注释标注引用内容起止位置与来源。

## 第一部分：选择题（20分）

说明：本部分共4题，每题5分，题型为非定向选择（可能有一个或多个选项正确）。全对得5分，部分正确得2分，若选择了错误的选项则不得分

### 1-1 下列数组的定义方式中正确的是：

- A: `int 1a[10];`
- B: `int _a[10] = {0};`
- C: `int a1[] = {1, 2, 3};`
- D: `int n = 10, b[n] = {0};`
- E: `int c[10] = {1,2};`

答案：BCE

解析：

- A 项数组命名错误：数字不能被放在变量名的第一位
- D 项数组长度不能使用变量定义

### 1-2 以下代码中语句 `i++` 被执行的次数为：

```
n = 1;
do
{
    n++;
    if(n == 4)
    {
        break;
    }
    i++;
}
while(n <= 4);
```

- A: 1
- B: 2
- C: 3
- D: 4
- E: 不确定

答案：B

解析：

当  $n = 2, 3$  时执行了  $i++$

### 1-3 以下给字符数组 `str` 的定义和赋值正确的是：

- A: `char str[10]; str = "China";`
- B: `char str[] = "China";`
- C: `char str[10]; strcpy (str, "abcdefghijklmn");`
- D: `char str[10] = "abcdefghijklmn";`
- E: `char str[10] = "china"`

答案：BE

解析：

- A 项字符串赋值要用 `strcpy()` 函数，不能使用等号
- C 项 `strcpy()` 函数中复制的字符串长度超过了字符串数组的长度
- D 项字符串初始化时的长度大于数组长度

### 1-4 下列说法中正确的是：

- A: 冒泡排序与选择排序的时间复杂度相同，都是  $O(n)$
- B: 字符串的长度等于其从左到右第一个出现的 `'\0'` 字符的下标值
- C: 对于字符串定义 `char str[8] = "abcdefg"` 有 `sizeof(str) == strlen(str)`
- D: `continue` 语句的作用是跳过这个循环，继续执行循环后的语句
- E: `do{ [A] }while();` 语句中的 `[A]` 部分至少会被执行一次

答案：BE

解析：

- A 项两种排序方法的时间复杂度均为  $O(n^2)$
- C 项 `sizeof(str) = 8, strlen(str) = 7` 不相等
- D 项 `continue` 的作用是跳过本次循环，继续执行本循环的下一次操作

## 第二部分：填空题（15分）

说明：本部分共2题6空，每题相应的得分标准见题目。

### 2-1 （每空 3 分）

在 ——（头文件名）中的 —— 函数具有连接两个字符串的功能，—— 函数具有比较两个字符串的功能。

答案：string.h strcat() strcmp()

### 2-2 （每空 2 分）

在一个有  $n$  个元素的数组中，其最后一个元素的下标是 ——，第一个元素的下标是 ——，已经声明过的数组的长度是 ——（可以/不可）变化的。

答案：n-1 0 不可

## 第三部分：简答题（15分）

说明：本部分共2题，每题相应的得分标准见题目。

### 3-1 （5 分）

说明以下用于将 `num` 数组中的 `n` 个有效数字从大到小排序的代码中的问题并说明原因：

```
int num[100],i,j,max;
for(i=0;i<n-1;i++)
{
    max = 0;
    for(j=i;j<n-1;j++)
    {
        if(num[j] > num[j+1])
        {
            max = j;
        }
    }
    int temp=num[max];
    num[max] = num[i];
    num[i] = temp;
}
```

答案：第四行应为 `max = i;` 理由：排序算法每次循环都使无序的部分向右移动一位，若 `max` 的值每次都从 `0` 开始取则会导致每次寻找到的最大项恒为整个数组中的最大元素

### 3-2 请写出与标准库函数 `strcpy` 的主要功能相同的函数（10分）

```
strcpy(char dest[], char src[])
{
}
}
```

答案：见下（不唯一）

```
strcpy(char dest[], char src[])
{
    int i = 0;
    do
    {
        dest[i] = src[i];
    } while(src[i] != 0);
}
```

## 第四部分：代码题（50分）

说明：本题共两大题，第一大题30分，第二大题20分，其中第二大题难度较高，请合理安排时间。详细评分规则见题目的数据要求

### 4-1 平均数和中位数

Nio Ku 获得了一堆数字，她想知道这些数字的中位数和平均数分别是多少

输入：两行，第一行为一个代表数字个数的数字  $n$ ；第二行为 Nio Ku 所获得的  $n$  个数字

输出：两行，分别输出上述第二行数字的算术平均数（保留一位小数）与中位数（从小到大排序后取第  $n/2$ （向上取整）位）

样例输入 1:

```
3
1 2 6
```

样例输出 1:

```
3.0
2
```

样例输入 2:

```
5
4 8 10 100 9
```

样例输出 2:

```
26.2
9
```

数据说明:

- $n < 1000$
- 输入的数字均在 `int` 范围内
- 共 10 组测试数据，每组满分为 3 分
- 对于每组数据，若仅平均数或中位数输出正确则得 2 分，都正确得满分

答案：见 `AvgMid.c`

解析:

知识点：排序，数组

难点：排序算法

易错点：虽然输入的数字在 `int` 范围内，但其总和和可能在 `int` 范围外，应该使用 `long long` 存储总和以防止溢出，减小浮点误差

## 4-2 表达式计算（仅加减）

APD 的算数不是太好，连加减运算都有可能出错，请你编写程序帮 APD 计算一段表达式的值

输入：一行一个仅含加减运算符和数字的表达式，以等号“=”结尾，其中可能有复数个空格

输出：一行一个数字，为表达式的值

样例输入 1:

```
1+1+1=
```

样例输出 1:

3

样例输入 2:

1 +22 + 650- 7 =

样例输出 2:

666

数据说明:

- 表达式长度不超过 1000
- 表达式中的数字与表达式的结果均在 `int` 范围内
- 共 10 组测试数据，每组 2 分
- 对于 40% 的数据，输入中不含空格
- 对于 40% 的数据，输入中每个运算符前后有一个空格
- 对于 20% 的数据，输入中空格随机出现

相关知识:

从标准输入中读入完整一行字符可以使用 `stdio.h` 中的 `gets()` 函数

[相关链接](#)

答案：见 `calc_1d.c`

解析:

知识点：循环，字符串处理

难点：忽略空格，读入

易错点：字符串转数字出错