

新思路 2024 年秋季大一招新机试试题<C 语言>

一、选择题（每题 4 分，共 20 分）

1. <从这里开始你的编程之路吧！>

下面四个选项中，均是不正确的八进制数或十六进制数的选项是：

- A. 077 0x1F 012
- B. 089 0xG2 019
- C. 014 0x2A 0x11
- D. 0xAB 075 0xFF

2. <选择题四个选项你都选不到正确的，还想在十四亿人中找到正确的人...>

下列说法不正确的是：

- A. C 语言中的函数可以没有返回值
- B. 全局变量在整个程序运行期间都占用内存
- C. 静态局部变量在函数调用结束后内存会被释放
- D. 局部变量在函数调用时才分配内存，调用结束后释放

3. <“我每天带给你惊喜和希望，思念就像动态数组无限延展.....”>

以下不正确的定义语句是：

- A. `char name[] = "John";`
- B. `int b[4] = {1, 2, 3, 4};`
- C. `double d[] = {1.0, 2.0, 3.0, 4.0};`
- D. `char str[5] = "hello";`

4. <开学你对 C 语言爱搭不理，今天机试让你哭的飞起>

下面语句的输出结果是：

```
float calculate(float x, float y)

{ return (x * y + 1); }
```

```
int main() {  
  
    int a = 2, b = 3;  
  
    float result;  
  
    result = calculate((float)a / b, (float)(a + b));  
  
    printf("%.2f\n", result);  
  
    return 0;  
  
}
```

- A. 4.67
- B. 4.33
- C. 5.33
- D. 5.00

5.<试题就像蓝天白云，晴空万里，忽然一道难题>

假设我们有以下结构体定义：

```
struct employee {  
    char name[30];  
    int age;  
    float salary;  
} staff[4], *ptr = staff;
```

关于这个结构体的定义，下列说法正确的是：

- A. 如果将 age 声明为 float，age 所占的空间会变大。
- B. 可以通过 printf("%d", (staff + 1).age); 输出第一个员工的年龄。
- C. [] 本质上是一种运算符，但通过 [] 做出的修改会直接反映到对应的内存。
- D. 可以通过 printf("%s", (ptr + 2).name); 输出第三个员工的名字。

二、程序设计题（前两题一题 10 分，后四题一题 15 分，共 80 分）

1. <写完选择题有没有感觉到来自出题的学长对你们的关怀呢?>

输入底面半径 r 和 R 以及高 h ，计算圆台的表面积和体积 ($\text{PI} = 3.1415$)，保留四位小数输出。

可能用到的公式：

表面积公式为： $\pi * (r^2 + R^2 + L * (r + R))$ ，

体积公式为： $V = (1/3) \pi h (r^2 + rR + R^2)$

圆台的斜高 L : $((r - R)^2 + h^2)^{(1/2)}$

<math.h> 中 sqrt 为开方函数

其中 r 、 R 分别是上下底面半径， h 是高。

输入： $r \quad R \quad h$

输出：表面积 体积

输入示例：

```
1 32.5 40.6 20.5
```

输出示例：

```
1 13558.3910 86385.2340
```

2. <这题给你们放放水，写个小学题！你不会写不出来吧？不会吧不会吧？？？>

编写一个程序，输出从 1000 年到 2024 年(包含二者)间所有的闰年总数，同时输出所有闰年年份之和。

闰年判断：年份是 4 的倍数，且不是 100 的倍数的。或年份是整百数的，且是 400 的倍数的。

输出：

```
1 249 376884
```

3. <确认过眼神，你是不会这道编程题的人>

编写一个程序：输入 L 与 R （均为正整数）。输出 $[L, R]$ 范围内的所有整数中，数字 2 出现的次数。

输入:

```
1 | 2 22
```

输出:

```
1 | 6
```

4.<让你掉下眼泪的不只是昨夜的酒，还有今天的机试>

编写一个程序：实现输入 10 个仅有一位小数的浮点数，并将其从大到小排序后保留两位小数输出。数之间用空格隔开，最后一行输出换行且无多余空格。

输入:

```
1 | 3.5 2.1 4.6 1.2 5.0 3.3 2.8 4.0 0.9 2.5
```

输出:

```
1 | 5.00 4.60 4.00 3.50 3.30 2.80 2.50 2.10 1.20 0.90
```

5.<题目略长，不要彷徨，套路不深，何必当真>

有一群孩子和一堆饼干（孩子数和饼干数均不超过 100），每个孩子有一个饥饿度，每个饼干都有一个大小。每个孩子只能吃一个饼干，且只有饼干的大小不小于孩子的饥饿度时，这个孩子才能吃饱。

编写一个程序，实现下述要求。

输入：孩子数量和每个孩子的饥饿度，饼干数量和每个饼干的大小。（二者数量可能不相同）

输出：最多有多少孩子可以吃饱。

输入:

```
1 | 2
2 | 1 2
3 | 3
4 | 1 2 3
```

输出:

```
1 | 2
```

输入:

```
1 | 3
2 | 1 2 3
3 | 2
4 | 1 1
```

输出:

```
1 | 1
```

6.<有些歌第一句就喜欢了，有的题看一眼就不会了>

将 1 到 1000 之间所有既是 3 的倍数又是回文数的数，通过 C/C++ 文件输入/输出方法输出到 outputs.txt 文件中。要求为保存为如下的形式：每五个数字一行中间间隔为一个空格。
(回文数是指像 14641 这样“对称”的数，即数的“倒序数”和原数一样。)
若采用手动输入酌情给分。

输出：

```
1 33 66 99 363 393
2 696 969
```