



## RC.CODE.C++ 赛项说明

### 1 赛项信息

#### 1.1 赛项内容

- A. 赛项将全面检验选手基于 C++ 软件编程语言的技术实现能力，鼓励创新、培养实践能力和解决问题的能力。
- B. 赛项均为个人参与。
- C. 赛项题目设置均为客观题+编程题。
- D. 赛项各组别均为使用电脑在规定平台参与答题。
- E. 赛项时长均为 90 分钟。

#### 1.2 组别设置

编程项目	编程语言	组别划分	年龄要求
C++编程	C++ 编程语言	C++ - A 组	≤12 岁
	C++ 编程语言	C++ - B 组	≤18 岁

#### 1.3 赛项题目

项目组别	单选题	编程题	项目时间
C++ - A 组	5 道题，共 100 分	5 道题，共 300 分	90 分钟
C++ - B 组	5 道题，共 100 分	5 道题，共 300 分	90 分钟

## 2 规则和得分

### 2.1 比赛规则

- A. 比赛要求选手在规定平台使用编码的方式，完成指定题目作答，答题过程中禁止切出编译器，禁止打开其他软件和网页，否则一律视为作弊，取消成绩。
- B. 每位选手只有一次挑战机会，规定时间未登录规定平台的选手视为放弃挑战。
- C. 准备阶段，选手须按照指定平台规定步骤完成作答准备，并自主操作进入正式答题。
- D. 挑战期间，选手不得离开电脑答题区域。
- E. 挑战期间，选手不得抄袭他人、不得作弊、不得直接与其他选手的电脑直接接触，如有违反，该选手记 0 分。
- F. 挑战过程中，不得求助他人，不得干扰其他选手备赛和答题，不得损坏公用设备，一经发现，勒令退出。
- G. 挑战期间，电脑上不得开启任何通讯软件，如有违反，该选手记 0 分。
- H. 挑战期间，选手不得使用手机、电话手表等通信、摄影电子设备及外接存储设备，如



有违反，取消挑战资格。

- I. 本规则的解释权归 RC 组委会所有。

## 2.2 赛项得分

根据题目完成情况以及完成度和时间综合评定，完成题目数量越多、完成度越高且用时较短的选手成绩越高。

## 3 设备要求

- A. 自备电脑，电脑操作系统：Mac OS、Windows 10 或以上操作系统；浏览器采用谷歌浏览器（69.0 版本以上）、Firefox、Internet Explorer 11 以上，推荐 Chrome。
- B. 自备智能手机一部。

## 4 大纲要求

### 4.1 C++ - A 组大纲

- A. 程序基础：
- a. 顺序结构：理解程序流程、基本输入输出。
    - a) 掌握变量的定义与使用。
    - b) 掌握基本运算语句和输入输出语句。
    - c) 能够完成顺序结构的程序。
  - b. 分支结构：if 条件句、简单逻辑运算。
    - a) 理解逻辑运算的基本概念。
    - b) 掌握基本逻辑表达式的功能及写法。
    - c) 能够完成选择结构的程序。
    - d) 掌握三目运算符的写法。
  - c. 循环结构：for 循环、while 循环来解决重复任务。
    - a) 掌握基本循环语句的功能及写法。
    - b) 能够完成循环结构的程序。
  - d. 数组：使用数组存储和访问数据集合。
    - a) 理解数组的概念。
    - b) 掌握数组的定义、赋值和查询方法。
  - e. 字符串：字符串操作基础，如连接、搜索字符等。
    - a) 掌握字符串数组相关函数的使用方法。
- B. 数理知识：
- a. 代数：整式加减乘除运算。
    - a) 了解常用数学函数并掌握其使用方法。
  - b. 几何：了解坐标系内点和线段表示方法。
    - a) 通过编程实现简单几何图形操作。
  - c. 函数：认识一次函数及其图像。



- a) 初步掌握数学库函数的使用。
- C. 算法:
  - a. 模拟：按照题目描述直接实现功能。
    - a) 理解模拟法的概念。
    - b) 能够使用模拟法解决较为简单的问题。
  - b. 枚举：使用循环穷举可能性来找到答案。
    - a) 理解枚举法的概念。
    - b) 能够使用枚举法解决较为简单的问题。

### 4.2 C++ - B 组大纲

- A. 程序基础：
  - a. 顺序结构：理解程序流程、基本输入输出。
    - a) 掌握基本的文件读写操作。
    - b) 掌握变量的类型转换。
  - b. 分支结构：if 条件句、简单逻辑运算。
    - a) 掌握多层分支结构的使用方法。
    - b) 能够编写多层分支结构的程序。
  - c. 循环结构：for 循环、while 循环来解决重复任务。
    - a) 掌握多层循环结构的使用方法。
    - b) 能够编写多层循环结构的程序。
  - d. 数组：使用数组存储和访问数据集合，多维数组进阶应用。
    - a) 能够完成包含分支结构、循环结构、数组等内容的综合问题。
  - e. 字符串：字符串操作基础，如连接、搜索字符等。字符串处理进阶，包括子串提取等高级操作。
    - a) 掌握字符数组相关函数的高级使用。
  - f. 结构体定义与使用。
    - a) 理解结构体和类的概念。
    - b) 掌握结构体和类的使用方法。
  - g. 分支结构与循环结构涉及更复杂逻辑判断与嵌套使用。
    - a) 结合二级 c、d、e、f 的要求，实现复杂逻辑。
  - h. 多关键字排序以及去重排序技巧。
    - a) 掌握桶排序、冒泡排序、选择排序、插入排序、快速排序、归并排序等多种排序方法。
  - i. 自定义函数以及递归调用概念强化；文件操作入门。
    - a) 理解函数的概念。
    - b) 掌握函数的定义与调用方法。
    - c) 理解函数参数传递的概念。
    - d) 掌握函数参数的传递方法。
    - e) 掌握递归的概念与方法。



- f) 掌握基本的文件读写操作。
- B. 数据结构：
  - a. set/map/pair: 掌握关联容器 set/map 以及数据对 pair。
    - a) 了解 STL 中的 set 和 map。
    - b) 掌握 STL 中 set 和 map 的使用方法。
  - b. 栈/队列: 使用标准库中 stack/queue 完成特定任务。
    - a) 掌握栈和队列的使用方法。
  - c. 链表: 基本链表节点创建与遍历。
    - a) 掌握各类链表（单向链表、双向链表、循环链表等）的使用方法。
- C. 数理知识：
  - a. 函数: 包括二次函数和反比例函数。
    - a) 通过编程实现函数图像绘制和计算。
  - b. 方程: 解二次方程以及方程应用问题。
    - a) 掌握扩展欧几里得算法并能利用其解同余方程。
    - b) 掌握高斯消元法并能利用其解决问题。
  - c. 组合计数: 初步了解排列组合概念。
    - a) 掌握加法原理、乘法原理并能解决简单的问题。
    - b) 理解排列与组合的定义。
    - c) 掌握排列与组合的计算公式。
- D. 算法：
  - a. 模拟: 按照题目描述直接实现功能。
    - a) 能够使用模拟法解决较为复杂的问题。
  - b. 枚举: 使用循环穷举可能性来找到答案。
    - a) 能够使用枚举法解决较为复杂的问题。
  - c. 高精度操作入门。
    - a) 掌握高精度的四则运算方法。
  - d. 分治思想应用。
    - a) 掌握二分法并能够使用其解决较为简单的问题。
    - b) 理解倍增法的思想并能够使用其解决问题。
  - e. 贪心算法简单应用。
    - a) 掌握简单的贪心算法并能够使用其解决较为简单的问题。
    - b) 能够设计较为复杂的贪心算法并用其解决问题。
  - f. 排序算法包含但不限于归并排序与快速排序。
    - a) 掌握快速排序、归并排序等排序方法。
  - g. 搜索算法及其剪枝。
    - a) 掌握深度优先搜索和广度优先搜索。
    - b) 掌握各类搜索剪枝方。
  - h. 动态规划算法。



- a) 理解动态规划的概念。
- b) 能够使用背包算法解决较为简单的问题。
- c) 能够使用区间动态规划解决较为简单的问题。
- d) 能够设计较为简单的动态规划并用其解决问题。
- e) 能够设计较为复杂的动态规划并用其解决问题。
- f) 掌握树形、状态压缩等复杂结构上的动态规划并利用其解决问题。
- g) 掌握复杂动态规划的状态设计方法并能利用其解决问题。
- h) 掌握斜率优化、决策单调性优化等复杂的动态规划优化方法并能利用其解决问题。
- i. 数论算法。
  - a) 掌握求最大公约数的欧几里得算法。
  - b) 掌握素数的埃氏筛法和线性筛法。
  - c) 掌握模运算的基本性质。
  - d) 理解逆元的概念。
  - e) 掌握费马小定理并能利用其求逆元。
  - f) 掌握欧拉定理并能利用其解决问题。
  - g) 掌握扩展欧拉定理并能利用其解决问题。
  - h) 掌握中国剩余定理并能利用其解同余方程组。
- j. 图论算法。
  - a) 掌握图的定义、存储与遍历。
  - b) 理解拓扑排序的思想并掌握有向无环图上的拓扑排序。
  - c) 掌握各类最短路算法。
  - d) 掌握各类最小生成树算法。
  - e) 理解连通分量、强连通图、强连通分量等概念。
  - f) 掌握有向图强连通分量的求法。
  - g) 掌握无向图割点、割边的求法。
  - h) 掌握无向图点双连通分量、边双连通分量的求法。
  - i) 掌握缩点的思想并利用其解决问题。
- k. 字符串算法。
  - a) 掌握字符串匹配的 KMP 算法。
- l. 高级数据结构。
  - a) 掌握 ST 表的使用。
  - b) 掌握并查集的使用。
  - c) 掌握树状数组的使用。
  - d) 掌握线段树的使用。
  - e) 掌握 trie 树的使用。
  - f) 掌握至少一种平衡树的使用，包括但不限于 treap、splay 等。
- m. 其他算法。



- a) 掌握快速幂算法。
- b) 掌握简单的递推算法并能够使用其解决较为简单的问题。
- c) 掌握前缀和算法并能够使用其解决较为简单的问题。
- d) 掌握差分算法并能够使用其解决较为简单的问题。
- e) 掌握双指针算法并能够使用其解决较为简单的问题。
- f) 理解哈希算法的概念并能设计较为简单的哈希函数。
- g) 掌握哈希表的用法。
- h) 理解欧拉回路的概念。
- i) 掌握求欧拉回路的方法。
- j) 掌握容斥原理并能利用其解决问题。
- k) 理解线性基的结构和基本性质并能利用其解决问题。

## 5 RC.CODE.C++ 样题示例

### 【C++ - A 组】样题示例

#### 一、选择题(每题 20 分)

1、下列表达式中正确的是( )

- A. 7++
- B. (a+b)++
- C. ++(a+b)
- D. ++x

答案 D

2、若定义 `int a=2, b=2`, 下列表达式中值为 6 的是( )

- A. `a*(++b)`
- B. `a*(b++)`
- C. `a+b`
- D. `a*b`

答案 A

3、在 `t` 为 `int` 类型, 进入下面的循环之前, `t` 的值为 0。则下列说法中正确的是( )

```
while(t=1){...}
```

- A. 循环条件表达式的值为 0
- B. 循环条件表达式的值为 1
- C. 循环条件表达式不合法
- D. 以上说法都不对

答案 B



4、下列 4 种不同进制表示的数中，数值最大的是（ ）

- A. 八进制数 52
- B. 十进制数 44
- C. 十六进制数 2F
- D. 二进制数 101000

**答案 C**

5、按照“先进后出”原则组织数据的结构是（ ） L = []

- A. 队列
- B. 栈
- C. 双向链表
- D. 二叉树

**答案 B**

## 二、编程题（总分 300 分）

### 1、编程题一（40 分）

**编程实现：排序**

**题目描述：**

输入三个正整数，将三个正整数按从小到大排序输出。

**输入描述：**

输入三个正整数，且正整数之间以一个空格分隔。

**输出描述：**

将三个正整数从小到大排序输出，且正整数之间以一个空格分隔。

**样例输入：**

15 8 10

**样例输出：**

8 10 15

**评分标准：**

10 分：能正确输出一组数据；

10 分：能正确输出两组数据；

10 分：能正确输出三组数据；

10 分：能正确输出四组数据。



## 2、编程题二 (40 分)

### 编程实现: 运费

#### 题目描述:

乘坐飞机时, 行李超出规定重量后, 会对行李进行托运且收取托运费。

以下是某航空公司行李托运的收费标准:

行李重量在 20 公斤内 (含 20) 按照每公斤 10 元收取费用, 超过 20 公斤的部分按照每公斤 15 元收取费用。

请编写程序, 在给定行李总重量 (单位: 公斤) 的情况下计算出托运行李的费用。

#### 输入描述:

输入一个正整数  $N$  ( $5 < N < 200$ ) 作为行李的总重量 (单位: 公斤)。

#### 输出描述:

输出托运  $N$  公斤行李的费用。

样例输入: 10

样例输出: 100

### 评分标准:

10 分: 能正确输出一组数据;

10 分: 能正确输出两组数据;

10 分: 能正确输出三组数据;

10 分: 能正确输出四组数据。

## 3、编程题三 (60 分)

### 编程实现: 求质数和

#### 题目描述:

给定一个正整数  $N$ , 计算出 2 到  $N$  之间所有质数的和。

例如: 如果  $N$  等于 10, 2 到  $N$  之间的质数有 2、3、5、7, 所有质数的和等于 17 ( $2 + 3 + 5 + 7$ )。

#### 输入描述:

输入一个正整数  $N$  ( $2 < N < 101$ )。

#### 输出描述:

输出一个整数, 表示 2 到  $N$  之间 (包含 2 和  $N$ ) 所有质数的和。





样例输入：

10

样例输出：

17

**评分标准：**

15 分：能正确输出一组数据；

15 分：能正确输出两组数据；

15 分：能正确输出三组数据；

15 分：能正确输出四组数据。

#### 4、编程题四 (60 分)

**编程实现：求降序子序列长度**

**题目描述：**

有一组长度为  $N$  的整数序列，从序列中找出包含第  $K$  个整数的最长降序子序列，并输出子序列的长度（降序子序列是指序列中的数字从左到右为递减状态，即右边的数字小于左边，相等不算递减）。

例如：长度为 5 的整数序列[4, 6, 2, 4, 8]， $K$  为 2。

第 2 个整数为 6，包含 6 的最长降序子序列为[6, 4, 2]，故长度为 3。

**输入描述：**

第一行输入两个正整数  $N$  ( $2 < N < 100$ )， $K$  ( $0 < K \leq N$ )，分别表示整数序列长度和需要包含的第几个元素。

第二行输入  $N$  个整数序列 ( $-1000 < \text{整数} < 1000$ )，整数之间以一个空格隔开。

**输出描述：**

输出包含第  $K$  个元素的最长的降序子序列的长度。

样例输入：

5 2

4 6 2 4 8

样例输出：

3



## 评分标准:

15 分: 能正确输出一组数据;

15 分: 能正确输出两组数据;

15 分: 能正确输出三组数据;

15 分: 能正确输出四组数据。

## 5、编程题五 (100 分)

### 编程实现: 最小值

#### 题目描述:

给定一个只包含数字的字符串, 长度为  $N$  ( $5 < N \leq 20$ ), 和一个正整数  $M$  ( $1 \leq M \leq 5$ )。使用  $M$  个乘号插入到字符串中, 且两个乘号不能相邻。插入后, 生成一个乘法算式。找出一种使乘法算式数值最小的插入方式, 并将结果输出。(乘号不能放在字符串的首尾位置)

#### 注意:

插入位置: 乘号可以插入在数字之间, 但不能放在字符串的首尾位置, 也就是说, 乘号只能放在两个数字之间。

相邻乘号: 插入的乘号不能相邻, 这意味着在两个乘号之间必须有至少一个数字。

例如, 字符串为 123456, 插入 2 个乘号。

插入方式有:

$$1 \times 2 \times 3456 = 6912$$

$$1 \times 23 \times 456 = 10488$$

$$1 \times 234 \times 56 = 13104$$

$$1 \times 2345 \times 6 = 14070$$

$$12 \times 3 \times 456 = 16416$$

$$12 \times 34 \times 56 = 22848$$

$$12 \times 345 \times 6 = 24840$$

$$123 \times 4 \times 56 = 27552$$

$$123 \times 45 \times 6 = 33210$$

$$1234 \times 5 \times 6 = 37020$$

其中乘法算式数值最小的是第二个, 为 6912。

#### 输入描述:

第一行输入两个正整数  $N$  ( $5 < N \leq 20$ ),  $M$  ( $1 \leq M \leq 5$ ),  $N$  表示字符串长度,  $M$  代表乘号个数, 两个正整数之间一个空格隔开。

第二行输入一个长度为  $N$ , 且只包含数字的字符串, 表示要插入  $M$  个乘号的字符串。

#### 输出描述:

输出一个整数, 表示最小的结果。



样例输入：

6 2

123456

样例输出：

6912

**评分标准：**

25 分：能正确输出一组数据；

25 分：能正确输出两组数据；

25 分：能正确输出三组数据；

25 分：能正确输出四组数据。

### 【C++ - B 组】样题示例

#### 一、选择题（每题 20 分，总分 100 分）

1、C++的基本数据类型不包括以下哪一项？（ ）

- A. int
- B. float
- C. string
- D. char

**答案 C**

2、在 C++中，哪个符号用于表示注释？（ ）

- A. //
- B. /\* \*/
- C. #
- D. A 和 B 都正确

**答案 D**

3、以下哪个选项是 C++中用于输出的标准库函数？（ ）

- A. input()
- B. print()
- C. cout
- D. output()

**答案 C**



4、以下关于 C++ 结构体的说法，正确的是（ ）

- A. 结构体中只能包含成员变量，不能包含成员函数
- B. 结构体不能从另一个结构体继承
- C. 结构体里面可以包含静态成员变量
- D. 结构体里面不能包含构造函数

答案 C

5、执行以下代码，输出的结果是（ ）

```
#include <iostream>
using namespace std;
int f(int k)
{
    if (k <= 2)
    {
        return 1;
    }
    return 2 * f(k - 2) + f(k - 1);
}
int main()
{
    int n = 7;
    cout << f(n);
    return 0;
}
```

- A. 21
- B. 41
- C. 43
- D. 45

答案 C

二、编程题（总分 300 分）

1、编程题一（40 分）

**编程实现：求位数**

**题目描述：**

给定一个正整数 N ( $1 < N < 200$ )，输出 N 为几位数。



**输入描述：**

输入一个正整数  $N$  ( $1 < N < 200$ )。

**输出描述：**

输出一个整数，表示  $N$  为几位数。

**样例输入：** 11

**样例输出：** 2

**评分标准：**

10 分：能正确输出一组数据；

10 分：能正确输出两组数据；

10 分：能正确输出三组数据；

10 分：能正确输出四组数据。

## 2、编程题二 (40 分)

**编程实现：拼写单词**

**题目描述：**

四种水果的英文单词，分别为 Apple、Banana、Cherry、Date。老师通过提示每个单词的首字母，让同学将对应的单词拼写下来。

请编写程序：

当输入的大写字母为 "A" 时，则输出 "Apple"；

当输入的大写字母为 "B" 时，则输出 "Banana"；

当输入的大写字母为 "C" 时，则输出 "Cherry"；

当输入的大写字母为 "D" 时，则输出 "Date"。

**输入描述：**

输入 A、B、C、D 中任意一个大写字母。

**输出描述：**

输出一个字符串，表示大写字母对应的英文单词（单词首字母大写）。

**样例输入：** A

**样例输出：** Apple



**评分标准：**

10分：能正确输出一组数据；

10分：能正确输出两组数据；

10分：能正确输出三组数据；

10分：能正确输出四组数据。

**3、编程题三（60分）**

**编程实现：判断数字**

**题目描述：**

给定一个正整数  $N$  ( $100 \leq N < 100000$ )，统计出 100 到  $N$ （包含 100 和  $N$ ）之间的正整数中，有多少个正整数满足以下条件：

- 1) 正整数个位数不为 3；
- 2) 正整数十位数不为 5；
- 3) 正整数百位数不为 7。

**输入描述：**

输入一个正整数  $N$  ( $100 \leq N < 100000$ )。

**输出描述：**

输出 100 到  $N$ （包含 100 和  $N$ ）之间有多少个正整数满足条件。

**样例输入：** 110

**样例输出：** 10

**评分标准：**

15分：能正确输出一组数据；

15分：能正确输出两组数据；

15分：能正确输出三组数据；

15分：能正确输出四组数据。



### 4、编程题四 (60 分)

#### 编程实现：计算 24

##### 题目描述：

“计算 24” 是一个经典数字游戏。

游戏规则是：从 1~10 之间的自然数任意拿出 4 个数（4 个数各不相同，顺序随机），进行加、减、乘三种运算（使用某种运算的次数、种类不限），要求运算结果等于 24。乘法的优先级高于加、减，并且算式中不可以用括号，数字的顺序必须保持不变。

##### 示例：

若给出的 4 个数是：10、2、4、8，则有两种解答方案 ( $10+2+4+8=24$ ,  $10*2-4+8=24$ )，则输出 2。

若给出的 4 个数是：7、2、3、6，则有零种解答方案，则输出 0。

##### 输入描述：

输入四个 1 到 10 之间的正整数并以一个空格隔开（四个正整数各不相同）。

##### 输出描述：

输出有多少种运算方案的结果为 24。

样例输入：10 2 4 8

样例输出：2

#### 评分标准：

15 分：能正确输出一组数据；

15 分：能正确输出两组数据；

15 分：能正确输出三组数据；

15 分：能正确输出四组数据。

### 5、编程题五 (100 分)

#### 编程实现：带约束的城市通行问题

##### 题目描述：

某地区有  $n$  个城市（编号 1 到  $n$ ）和  $m$  条双向公路。每条公路连接两个城市，并有一个通行时间（正整数）。部分城市设有“充电站”。

一辆电动车从起点  $s$  出发，目标是到达终点  $t$ 。电动车的电池容量为  $C$ （即满电状态下最多能行驶  $C$  单位时间），初始时电池为满电。

通行规则：



每经过一条通行时间为  $w$  的公路，电池电量会减少  $w$

若到达一个充电站，可选择将电池充满（也可选择不充）

请找到一条从  $s$  到  $t$  的路径，使得总通行时间最短。若不存在可行路径，输出  $-1$ 。

### 输入描述：

第一行：五个整数  $n, m, s, t, C$  ( $2 \leq n \leq 50, 1 \leq m \leq 200, 1 \leq s, t \leq n, s \neq t, 1 \leq C \leq 100$ )

第二行： $n$  个整数 (0 或 1)，表示  $is\_charge[1]$  到  $is\_charge[n]$

接下来  $m$  行：每行三个整数  $u, v, w$ ，表示  $u$  和  $v$  之间有一条通行时间为  $w$  的公路  
( $1 \leq w \leq C$ )

### 输出格式

一行一个整数，表示最短总通行时间；若无法到达，输出  $-1$ 。

### 样例输入：

```
5 5 1 5 10
0 1 0 1 0
1 2 5
2 3 3
3 4 4
4 5 2
2 4 6
```

### 样例输出：

```
13
```

### 评分标准：

25 分：能正确输出一组数据；

25 分：能正确输出两组数据；

25 分：能正确输出三组数据；

25 分：能正确输出四组数据。