

GESP C++三级样题卷

(满分：100分 考试时间：90分钟)

学校：\_\_\_\_\_

姓名：\_\_\_\_\_

题目	一	二	三	总分
得分				

一、单选题 (每题 2 分, 共 30 分)

题号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
答案	A	A	D	C	A	B	B	A	D	B	C	C	B	D	C

1. 下列关于负数的原码、反码、补码的描述中，正确的是( )
  - A. 原码和反码互为按位取反 (符号位除外)，补码为反码加 1
  - B. 原码和反码互为按位取反 (符号位除外)，补码为原码加 1
  - C. 反码和补码互为按位取反 (符号位除外)，原码为反码加 1
  - D. 补码和原码互为按位取反 (符号位除外)，反码为补码加 1
2. 一个 `int` 类型的值乘以 8，等价于以下哪个位运算？( )
  - A. 左移 3 位
  - B. 右移 3 位
  - C. 左移 8 位
  - D. 右移 8 位
3. 以下哪个属于 C++语言中的位运算符？( )
  - A. +
  - B. -
  - C. \*
  - D. &

4. 若有以下代码，则数组 `arr` 的长度是( )

```
int arr[] = {1, 2, 3, 4, 5};
```

- A. 3
  - B. 4
  - C. 5
  - D. 6
5. 在 C++语言中，可以定义一个一维整型数组的是( )
- A. `int array[5];`
  - B. `int array[];`
  - C. `int[5] array;`
  - D. `int[] array;`
6. 枚举算法的主要特点是( )
- A. 以空间换时间
  - B. 逐个尝试所有可能的解
  - C. 动态规划
  - D. 贪心策略
7. 对于一个十进制数 37，以下哪个是它的二进制表示( )
- A. 10101
  - B. 100101
  - C. 101001
  - D. 1000101
8. 下列关于十六进制的描述中，正确的是( )
- A. 使用 0-9 和 A-F 表示
  - B. 使用 0-9 和 A-E 表示

- C. 使用 1-9 和 A-F 表示
- D. 使用 1-9 和 A-E 表示

9. 下列哪个是 C++语言中用于获取字符串长度的函数( )

- A. `length()`
- B. `len()`
- C. `getLength()`
- D. `strlen()`

10. 通常用下列哪种方式来描述算法?

- A. 汇编语言
- B. 伪代码
- C. SQL
- D. CSS

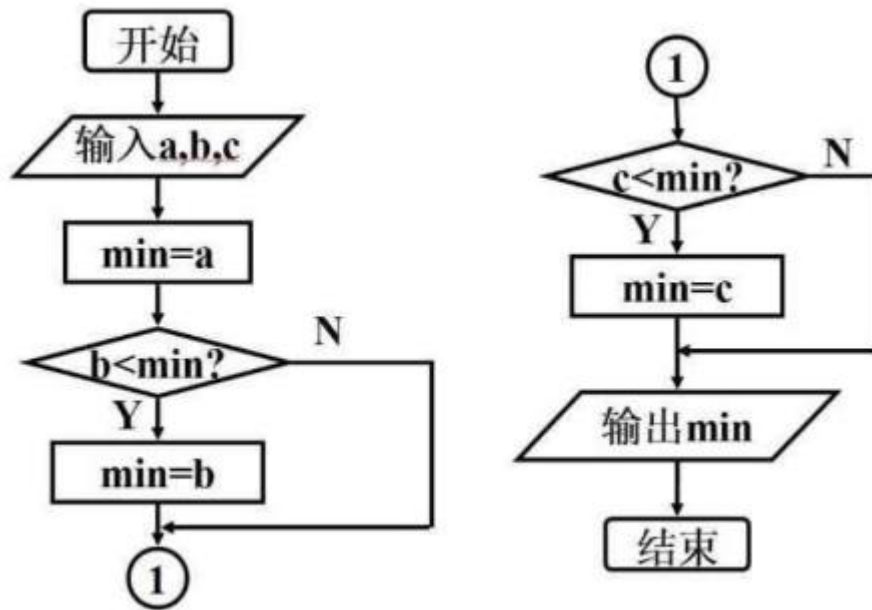
11. 如果 **a** 和 **b** 均为 `int` 类型的变量, 下列表达式能正确判断 “**a** 等于 0 且 **b** 等于 0” 的是 ( )

- A. `((~a) && (~b))`
- B. `((a & b) == 0)`
- C. `((a | b) == 0)`
- D. `((a ^ b) == 0)`

12. 如果 **a** 为 `int` 类型的变量, 下列哪个表达式可以正确求出满足 “大于等于 **a** 且是 4 的倍数” 的整数中最小的?

- A. `(a & (~3))`
- B. `(a / 4 * 4)`
- C. `((a - 1) | 3) + 1`
- D. `(a << 2)`

13. 下面流程图，输入 1 2 3，会输出( )



- A. 无输出
- B. 1
- C. 2
- D. 3

14. 在下列代码的横线处填写 ( )，可以保证输出是“1357”，不会有多余字符。

```
#include <iostream>
#include <string>
using namespace std;
int main() {
    char str[] = "1234567";
    for ( _____ ) // 在此处填入代码
        cout << str[i];
    return 0;
}
```

- A. `int i = 0; i < strlen(str); i++`
- B. `int i = 0; str[i] != '\0'; i++`

- C. `int i = 1; i <= 7; i += 2`
- D. `int i = 0; i <= 6; i += 2`

15. 在下列代码的横线处填写 ( ), 可以使得输出是“17 11”。

```
#include <iostream>
using namespace std;
int main() {
    int a = 11, b = 17;
    a = _____; // 在此处填入代码
    b = a ^ b;
    a = a ^ b;
    cout << a << " " << b << endl;
    return 0;
}
```

- A. `a + b`
- B. `a - b`
- C. `a ^ b`
- D. `a & b`

二、判断题 (每题 2 分, 共 20 分)

题号	1	2	3	4	5	6	7	8	9	10
答案	T	F	T	F	T	F	F	T	T	F

1. 二进制数据编码中, 负数的补码是通过将原码按位取反并加 1 得到的。 ( )
2. 在 C++ 语言中, 数组的下标从 1 开始计数。 ( )
3. 在 C++ 语言中, 字符串是以 '\0' 结尾的字符数组。 ( )
4. 在 C++ 语言中, 可以使用浮点数 (如 3.0) 作为数组下标。 ( )
5. 枚举算法是一种暴力求解方法, 逐个尝试所有可能的解。 ( )
6. C++ 语言中数字的符号位是不参与位运算的。 ( )
7. C++ 语言中的数组可以根据需要自动调整大小。 ( )
8. 在 C++ 语言中, 表达式 `(0xff == 255)` 的值为 `true`。 ( )
9. 如果 `a` 为 `int` 类型的变量, 且表达式 `((a & 1) == 0)` 的值为 `true`, 则说明 `a` 是偶数。 ( )
10. 表达式 `(7 >> 2)` 的计算结果为 1.75, 且结果类型为 `double`。 ( )

三、编程题 (每题 25 分, 共 50 分)

题号	1	2
答案		

1. 逛商场

小明是个不太有计划的孩子。这不,刚到手的零花钱,就全部拿着逛商场去了。小明的原则很简单,只要见到想买的物品而且能买得起,就一定会买下来之后才会继续往前走。一天下来,小明到底买了多少物品呢?

**【输入格式】**

输入共 3 行:

第一行是一个整数  $N$ , 表示商场中共有  $N$  种小明想买的物品 ( $1 \leq N \leq 100$ );

第二行共有  $N$  个整数, 分别表示小明先后见到想买的物品的价格;

第三行是一个整数  $X$ , 表示开始时小明共有  $X$  元零花钱。

**【输出格式】**

输出 1 行, 包含一个整数, 表示小明买到的物品数。

**【样例输入】**

```
6
7 5 9 10 7 4
30
```

**【样例输出】**

```
4
```

【参考代码】

```
#include <bits/stdc++.h>
using namespace std;
int price[100];
int main() {
    int n = 0, x = 0, cnt = 0;
    cin >> n;
    for (int i = 0; i < n; i++)
        cin >> price[i];
    cin >> x;
    for (int i = 0; i < n; i++) {
        if (x >= price[i]) { // 买得起就买
            x -= price[i];
            cnt++;
        }
    }
    cout << cnt;
    return 0;
}
```



## 2. 进制转换

### 【题目描述】

小美刚刚学习了十六进制，她觉得很有趣，想到是不是还有更大的进制呢？在十六进制中，用‘A’表示 10、‘F’表示 15。如果扩展到用‘Z’表示 35，岂不是可以表示 36 进制数了嘛！

### 【输入格式】

输入 2 行，第一行包含一个正整数 N，第二行包含一个正整数 R。

保证  $1 \leq N \leq 1000000$ ， $2 \leq R \leq 36$

### 【输出格式】

输出 1 行，为 N 的 R 进制表示。

### 【样例输入】

```
123
```

```
25
```

### 【样例输出】

```
4N
```

### 【参考代码】

```
#include <bits/stdc++.h>
using namespace std;
char res[30];
int len = 0;
char trans(int d) {
    if (d < 10)
        return (char)(d + '0');
    return (char)(d + 'A' - 10);
}
int main() {
    int n = 0, r = 0;
    cin >> n >> r;
    while (n > 0) {
        res[len] = trans(n % r);
        n /= r;
        len++;
    }
    for (int i = len - 1; i >= 0; i--)
        cout << res[i];
    cout << endl;
    return 0;
}
```