##### ***蒸汽创客***

##### 蓝桥杯第15届STEMA考试 C++ 试卷（2023年10月）

**一、选择题**

1) 十进制数189转换成二进制是（ A ）。 进制转换

A、10111101  
B、11010011  
C、11110000  
D、11001100

2）执行cout << '9'\*3;语句后，输出的结果是（  D ）。  
A   27

B    9\*3

C   999

D   171

3）下列选项中，判断a不等于0且b不等于0的正确的[条件表达式](https://so.csdn.net/so/search?q=%E6%9D%A1%E4%BB%B6%E8%A1%A8%E8%BE%BE%E5%BC%8F&spm=1001.2101.3001.7020" \t "/Users/xuzhiyuan/Documents\\x/_blank)是（ C ）。

A   !(a==0&&b==0)

B    !a=0&&!b=0

C   a && b

D   !((a!=0)&&(b!=0))

1. 下列选项中，按照数据类型的字节长度从小到大排序正确的是（ B ）。

A、char，long long，float

B、char，float，long long

C、long long，float，char

D、float，long long，char

5）以下选项中，哪一个是C++注释符号？（ B ）。

A、\*          B、//       C、;         D、#

6）执行cout<<(5%3);语句后，输出的结果是（ C ）。

A、0        B、1           C、2      D、3

7) 定义一个整型变量int a，使用指针存储a的地址，下列选项正确的是( D )

A、int &p = &a  
B、int \*\*p = &a  
C、int \*p = a  
D、int \*p = &a

8) 定义一个字符串数组：char a [1024] = "Welcome to China!"；  
要计算上述字符串的长度，应该使用下列哪个函数（ C ）。

A、strcat()  
B、strcmp()  
C、strlen ()  
D、size ()

9) 已定义变量int a = 10, b = 0；  
执行表达式((a = 5) || (b = 1))之后，a和b的值分别是（ C ）。

A、10 0  
B、5 1  
C、5 0  
D、10 1

10) 运行下面程序，输出的结果是（ B ）。

int func(int x, int y)

{

if(x == 1 && y == 1)

return 1;

if(x < 0)

return y;

if(y < 0)

return x;

return (func (x, y-2) + func (x-2, y));

}

int main()

{

cout << func (3,3);

return 0;

}

A、7  
B、8  
C、9  
D、10

**二、编程题**

6) 编程实现：与3有关的数  
提示信息：  
如果一个正整数的个位可以被3整除，则这个正整数与3有关。  
例如：  
10的个位是0，0能被3整除，故10与3有关；  
11的个位是1，1不能被3整除，故11与3无关；  
19的个位是9，9能被3整除，故19与3有关。  
  
题目描述：  
给定一个正整数N（1≤N≤10000），从小到大依次输出1到N之间所有与3有关的数  
输入描述：输入一个正整数N（1≤N≤10000）  
输出描述：输出一行，表示1到N之间（含1和N）所有与3有关的正整数，从小到大排列，正整数之间以一个空格隔开  
  
样例输入：10  
样例输出：3 6 9 10

#include <iostream>

**using** **namespace** std;

**int** main() {

**int** N;

    cin >> N;

**for** (**int** i = 1; i <= N; i++) {

**if** (i % 10 == 0 || i % 10 % 3 == 0) {

            cout << i << " ";

        }

    }

**return** 0;

}

1. 编程实现：跳水比赛  
   题目描述：  
   某校有N名选手参加跳水比赛。每名选手比赛后，由6名裁判评分，选手的最终得分为6名裁判评分的总和。  
   比赛结束之后，请计算出最高分与最低分相差多少。  
   例如：N = 4，表示有4名选手；  
   裁判对于这4名选手的打分是：  
   5 9 6 8 7 6  
   7 7 7 8 6 7  
   5 7 6 7 5 7  
   7 7 7 5 9 8  
   4名选手的最终分数分别为：41，42，37，43，其中最高分是43分，最低分是37分，那么最高分和最低分相差了6分，故输出6。  
     
   输入描述：  
   第一行输入一个正整数N（2≤N≤100），表示参赛选手的人数  
   接下来输入N行，每行6个整数（0≤整数≤10），表示每个裁判的评分，整数之间以一个空格隔开  
   输出描述：  
   输出一个整数，表示最高分与最低分的差值  
     
   样例输入：  
   4  
   5 9 6 8 7 6  
   7 7 7 8 6 7  
   5 7 6 7 5 7  
   7 7 7 5 9 8  
   样例输出：  
   6

#include <iostream>

**using** **namespace** std;

**int** main() {

**int** N;

    cin >> N;

**int** maxScore = INT\_MIN; // 初始化为最小整数

**int** minScore = INT\_MAX; // 初始化为最大整数

**for** (**int** i = 0; i < N; i++) {

**int** sum = 0; // 当前选手的得分总和

**for** (**int** j = 0; j < 6; j++) {

**int** score;

            cin >> score;

            sum += score;

        }

        // 更新最高分和最低分

**if** (sum > maxScore) maxScore = sum;

**if** (sum < minScore) minScore = sum;

    }

    // 输出最高分和最低分的差值

    cout << maxScore - minScore << endl;

**return** 0;

}

8) 编程实现：密文解密  
题目描述：  
有一段密文，加密规律如下：  
原文中所有的字符在字母表中的位置被左移了若干位（如：a左移一位后为z，A左移一位后为Z，f左移三位后为c......）。  
例如：密文为Ayz，已知加密时原文中的字母被左移了3位，解密后，原文是Dbc。  
请根据这个规律，对密文进行解密。  
  
输入描述：  
共两行  
第一行输入一个只包含大小写字母的字符串（长度小于100），表示密文  
第二行输入一个正整数n (1≤n≤100000)，表示原文中的字母在字母表中的位置被左移的位数  
输出描述：  
输出一个字符串，表示解密后的原文  
  
样例输入：  
Ayz  
3  
样例输出：  
Dbc

#include <iostream>

#include <string>

**using** **namespace** std;

**int** main() {

    string cipher;

**int** shift;

    // 读入密文和移位数

    cin >> cipher;

    cin >> shift;

    // 遍历密文中的每个字符进行解密

**for** (**int** i = 0; i < cipher.length(); ++i) {

**if** (isalpha(cipher[i])) {

**char** base = islower(cipher[i]) ? 'a' : 'A';

            // 计算解密后的字符位置

**int** pos = (cipher[i] - base + shift + 26) % 26;

            cipher[i] = base + pos;

        }

    }

    // 输出解密后的字符串

    cout << cipher << endl;

**return** 0;

}

1. 编程实现：翻转游戏币  
   题目描述：  
   桌面上有n枚游戏币，均为反面朝上，编号依次为1到n。有n个人（游戏币的数量与人的数量相等），首先第1个人将所有游戏币翻转，然后第2个人将所有编号是2的倍数的游戏币翻转，接下来第3个人将所有编号是3的倍数的游戏币翻转 ......以此类推，当最后一个人完成操作后，还有多少枚游戏币正面朝上？  
   例如：n = 4;  
   最初4枚游戏币的状态为：反反反反；  
   经过第1个人翻转后，游戏币的状态为：正正正正；  
   经过第2个人翻转后，游戏币的状态为：正反正反；  
   经过第3个人翻转后，游戏币的状态为：正反反反；  
   经过第4个人翻转后，游戏币的状态为：正反反正；  
   最后，还有两枚游戏币正面朝上，故答案为2。  
     
   输入描述：  
   输入一个正整数n（3≤n≤109），表示游戏币数量及人的数量  
   输出描述：  
   输出一个整数，表示最终游戏币正面朝上的数量  
     
   样例输入：  
   4  
   样例输出：  
   2

这个问题实际上是一个关于数学规律的问题。我们可以发现，只有当游戏币的编号有奇数个因数时，它才会正面朝上。只有完全平方数的编号才会有奇数个因数，因为除了平方根以外，其他因数都是成对出现的。

例如，对于编号4的游戏币（因数为1, 2, 4），只有在第1个人、第2个人和第4个人翻转时会被操作，共3次，因此它会以正面朝上结束。

因此，问题转化为了计算小于或等于n的完全平方数的数量。最简单的方法就是计算sqrt(n) 的整数部分，因为这表示了所有小于或等于n的完全平方数。

方法一：  
#include <iostream>

#include <cmath>

**using** **namespace** std;

**int** main() {

**long** **long** n;

    cin >> n;

    // 计算小于或等于n的完全平方数的数量

**int** count = **static\_cast**<**int**>(sqrt(n));

    cout << count << endl;

**return** 0;

}

方法二：

#include <iostream>

#include <stdlib.h>

#include <stdio.h>

#include <cstring>

**using** **namespace** std;

**int** main(){

**int** n;

    cin >> n;

**int** nums[n + 5];

    //全部关灯

    memset(nums, 0, **sizeof**(nums));

**for**(**int** i = 2; i <= n; i++){

**for**(**int** j = 1; j < 5000; j++){

**if**(i \* j <= n){

                nums[i \* j] = !nums[i \* j];

            }**else**

**break**;

        }

    }

**int** sum = 0;

**for**(**int** i = 1; i <= n; i++){

**if**(nums[i] == 1) sum++;

    }

    cout << sum << endl;

**return** 0;

}

1. 编程实现：分发糖果  
   题目描述：  
   n个学生站成一排，已知每名学生的考试成绩，老师要根据成绩按以下规则分发糖果：  
   1、每个学生至少得到一个糖果；  
   2、相邻两个学生中成绩高的会获得更多的糖果；  
   3、相邻两个学生成绩即使相同，获得的糖果数量也可以不同。  
   请计算出老师最少需要准备多少颗糖果？  
   例如：有3个学生，他们的考试成绩分别是70，50，80，可以给第一个学生2颗糖果，给第二个学生1颗糖果，给第三个学生2颗糖果，所以最少需要准备5颗糖果。  
     
   输入描述：  
   共两行  
   第一行输入一个正整数n（1≤n≤20000），表示学生人数  
   第二行输入n个整数（0≤整数≤100），表示每个学生的考试成绩，整数之间以一个空格隔开  
   输出描述：  
   输出一个整数，表示最少需要准备的糖果数量  
     
   样例输入：  
   3  
   70 50 80  
   样例输出：  
   5

#include <iostream>

**using** **namespace** std;

**int** main() {

**int** n;

    cin >> n;

**int** scores[n];

**int** candies[n];

**for** (**int** i = 0; i < n; ++i) {

        cin >> scores[i];

        candies[i] = 1; // 初始化，每个学生至少1颗糖果

    }

    // 从左到右遍历

**for** (**int** i = 1; i < n; ++i) {

**if** (scores[i] > scores[i - 1]) {

            candies[i] = candies[i - 1] + 1;

        }

    }

    // 从右到左遍历

**for** (**int** i = n - 2; i >= 0; --i) {

**if** (scores[i] > scores[i + 1] && candies[i] <= candies[i + 1]) {

            candies[i] = candies[i + 1] + 1;

        }

    }

    // 计算糖果总数

**int** totalCandies = 0;

**for** (**int** i = 0; i < n; ++i) {

        totalCandies += candies[i];

    }

    cout << totalCandies << endl;

**return** 0;

}

1. 编程实现：区间最小值  
   题目描述：  
   给定n个整数，从1到n顺序编号，接下来进行m次查询，第i次查询第ai个数到第bi个数（包含ai和bi）之间的最小值并输出。  
   例如：n = 8, 8个正整数依次为：40 20 10 30 70 50 80 60  
   m = 3，3次查询分别为：  
   a1 = 3, b1 = 7  
   a2 = 1, b2 = 2  
   a3 = 5, b3 = 8  
   第一次查询：第3个数（10）到第7个数（80）之间最小值是10；  
   第二次查询：第1个数（40）到第2个数（20）之间最小值是20；  
   第三次查询：第5个数（70）到第8个数（60）之间最小值是50；  
   故输出  
   10  
   20  
   50  
     
   输入描述：  
   第一行输入两个整数n和m (1≤n，m≤105 )，分别表示整数的数量及查询次数  
   第二行输入n个整数（0≤整数≤105）  
   接下来m行，每行输入2个整数ai和bi（1≤ai ≤bi ≤n），分别表示查询的起始位置和终止位置  
   输出描述：  
   输出共m行，每行输出一个整数，分别表示每次查询得到的第ai个数到第bi个数之间（包含ai和bi）的最小值  
     
   样例输入：  
   8 3  
   40 20 10 30 70 50 80 60  
   3 7  
   1 2  
   5 8  
   样例输出：  
   10  
   20  
   50

#include <iostream>

**using** **namespace** std;

**int** main() {

**int** n, m;

    cin >> n >> m;

**int** nums[n + 5];

**for** (**int** i = 0; i < n; ++i) {

        cin >> nums[i];

    }

**for** (**int** i = 0; i < m; ++i) {

**int** a, b;

        cin >> a >> b;

**int** minVal = nums[a - 1];

**for** (**int** j = a; j < b; ++j) {

**if** (nums[j] < minVal) {

                minVal = nums[j];

            }

        }

        cout << minVal << endl;

    }

**return** 0;

}