

2017年第二十三届NOIP信奥赛普及组初赛C++试题

题目总数：28 总分数：100

一、单项选择题

第 1 题 单选题

在8位二进制补码中，10101011表示的数是十进制下的（ ）。

- A. 43
- B. -85
- C. -43
- D. -84

答案 B

第 2 题 单选题

计算机存储数据的基本单位是（ ）。

- A. bit
- B. Byte
- C. GB
- D. KB

答案 B

第 3 题 单选题

下列协议中与电子邮件无关的是（ ）。

- A. POP3
- B. SMTP
- C. WTO
- D. IMAP

答案 C

第 4 题 单选题

分辨率为800x600、16位色的位图，存储图像信息所需的空间为（ ）。

- A. 937.5KB
- B. 4218.75KB
- C. 4320KB
- D. 2880KB

答案 A

第 5 题 单选题

计算机应用的最早领域是（ ）。

- A. 数值计算
- B. 人工智能
- C. 机器人
- D. 过程控制

答案 A

第 6 题 单选题

下列不属于面向对象程序设计语言的是（ ）。

- A. C
- B. C++
- C. Java
- D. C#

答案 A

第 7 题 单选题

NOI的中文意思是（ ）。

- A. 中国信息学联赛
- B. 全国青少年信息学奥林匹克竞赛
- C. 中国青少年信息学奥林匹克竞赛
- D. 中国计算机协会

答案 B

第 8 题 单选题

2017年10月1日是星期日，1999年10月1日是（ ）。

- A. 星期三
- B. 星期日
- C. 星期五

D. 星期二

答案 C

第 9 题 单选题

甲、乙、丙三位同学选修课程，从4门课程中，甲选修2门，乙、丙各选修3门，则不同的选修方案共有（ ）种。

- A. 36
- B. 48
- C. 96
- D. 192

答案 C

第 10 题 单选题

设G是有n个结点、m条边（ $n \leq m$ ）的连通图，必须删去G的（ ）条边，才能使得G变成一棵树。

- A. $m - n + 1$
- B. $m - n$
- C. $m + n + 1$
- D. $n - m + 1$

答案 A

第 11 题 单选题

对于给定的序列 $\{a_k\}$ ，我们把 (i, j) 称为逆序对当且仅当 $i < j$ 且 $a_i > a_j$ 。那么序列1, 7, 2, 3, 5, 4的逆序对数为（ ）个。

- A. 4
- B. 5
- C. 6
- D. 7

答案 B

第 12 题 单选题

表达式 $a * (b + c) * d$ 的后缀形式是（ ）。

- A. $a bcd * + *$
- B. $a bc + * d *$
- C. $a * bc + * d$
- D. $b + c * a * d$

答案 B

第 13 题 单选题

向一个栈顶指针为hs 的链式栈中插入一个指针 s 指向的结点时，应执行（ ）。

- A. $hs \rightarrow next = s;$
- B. $s \rightarrow next = hs; hs = s;$
- C. $s \rightarrow next = hs \rightarrow next; hs \rightarrow next = s;$
- D. $s \rightarrow next = hs; hs = hs \rightarrow next;$

答案 B

第 14 题 单选题

若串S="copyright"，其子串的个数是（ ）。

- A. 72
- B. 45
- C. 46
- D. 36

答案 C

第 15 题 单选题

十进制小数13.375对应的二进制数是（ ）。

- A. 1101.011
- B. 1011.011
- C. 1101.101
- D. 1010.01

答案 A

第 16 题 单选题

对于入栈顺序为 a, b, c, d, e, f, g 的序列，下列（ ）不可能是合法的出栈序列。

- A. a, b, c, d, e, f, g
- B. a, d, c, b, e, g, f
- C. a, d, b, c, g, f, e
- D. g, f, e, d, c, b, a

答案 C

第 17 题 单选题

设 A 和 B 是两个长为 n 的有序数组，现在需要将 A 和 B 合并成一个排好序 的数组，任何以元素比较作为基本运算的归并算法在最坏情况下至少要做（ ）次比较。

- A. n^2
- B. $n \log n$

- C. $2n$
- D. $2n-1$

答案 D

第 18 题 单选题

从 () 年开始, NOIP 竞赛将不再支持 Pascal 语言。

- A. 2020
- B. 2021
- C. 2022
- D. 2023

答案 C

第 19 题 单选题

一家四口人, 至少两个人生日属于同一月份的概率是 () (假定每个人生日属于每个月份的概率相同且不同人之间相互独立) 。

- A. $1/12$
- B. $1/144$
- C. $41/96$
- D. $3/4$

答案 C

第 20 题 单选题

以下和计算机领域密切相关的奖项是 () 。

- A. 奥斯卡奖
- B. 图灵奖
- C. 诺贝尔奖
- D. 普利策奖

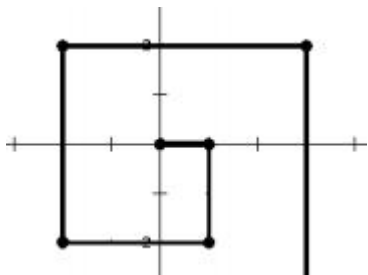
答案 B

二、问题求解

第 21 题 填空题

一个人站在坐标 (0,0) 处, 面朝x轴正方向。

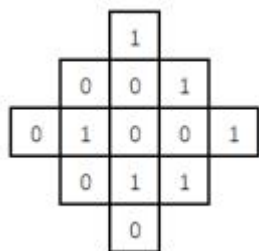
第一轮, 他向前走1单位距离, 然后右转; 第二轮, 他向前走2单位距离, 然后右转; 第三轮, 他向前走3单位距离, 然后右转.....他一直这么走下去。请问第2017轮后, 他的坐标是: (__, __)。(请在答题纸上用逗号隔开两空答案)



答案 1009, 1008

第 22 题 填空题

如右图所示，共有13个格子。对任何一个格子进行一次操作，会使得它自己以及与它上下左右相邻的格子中的数字改变（由1变0，或由0变1）。现在要使得所有的格子中的数字都变为0，至少需要_____次操作。



答案 3

三、阅读程序写结果

第 23 题 填空题

```

1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int t[256];
6      string s;
7      int i;
8      cin >> s;
9      for (i = 0; i < 256; i++)
10         t[i] = 0;
11     for (i = 0; i < s.length(); i++)
12         t[s[i]]++;
13     for (i = 0; i < s.length(); i++)
14         if (t[s[i]] == 1) {
15             cout << s[i] << endl;
16             return 0;
17         }
18     cout << "no" << endl;
19 
```

```
20 |     return 0;
    | }
```

输入: xyzxyw

输出: _____

答案 z

第 24 题 填空题

```
1 | #include <iostream>
2 | using namespace std;
3 |
4 | int g(int m, int n, int x) {
5 |     int ans = 0;
6 |     int i;
7 |     if (n == 1)
8 |         return 1;
9 |     for (i = x; i <= m / n; i++)
10 |         ans += g(m - i, n - 1, i);
11 |     return ans;
12 | }
13 |
14 | int main() {
15 |     int t, m, n;
16 |     cin >> m >> n;
17 |     cout << g(m, n, 0) << endl;
18 |     return 0;
19 | }
```

输入: 7 3

输出: _____

答案 8

第 25 题 填空题

```
1 | #include <iostream>
2 | using namespace std;
3 |
4 | int main() {
5 |     string ch;
6 |     int a[200];
7 |     int b[200];
8 |     int n, i, t, res;
9 |     cin >> ch;
```

```

10     n = ch.length();
11     for (i = 0; i < 200; i++)
12         b[i] = 0;
13     for (i = 1; i <= n; i++) {
14         a[i] = ch[i - 1] - '0';
15         b[i] = b[i - 1] + a[i];
16     }
17     res = b[n];
18     t = 0;
19     for (i = n; i > 0; i--) {
20         if (a[i] == 0)
21             t++;
22         if (b[i - 1] + t < res)
23             res = b[i - 1] + t;
24     }
25     cout << res << endl;
26     return 0;
27 }

```

输入：1001101011001101101011110001

输出：_____

答案 11

第 26 题 填空题

```

1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int n, m;
6      cin >> n >> m;
7      int x = 1;
8      int y = 1;
9      int dx = 1;
10     int dy = 1;
11     int cnt = 0;
12     while (cnt != 2) {
13         cnt = 0;
14         x = x + dx;
15         y = y + dy;
16         if (x == 1 || x == n) {
17             ++cnt;
18             dx = -dx;
19         }
20         if (y == 1 || y == m) {
21             ++cnt;
22             dy = -dy;

```



```

23     }
24     }
25     cout << x << " " << y << endl;
26     return 0;
27 }

```

输入 1: 4 3

输出 1: _____ (3 分)

输入 2: 2017 1014

输出 2: _____ (5 分)

答案 输出 1: 1 3
输出 2: 2017 1

四、完善程序

第 27 题 问答题

(快速幂) 请完善下面的程序, 该程序使用分治法求 $x^p \bmod m$ 的值。(第一空 2 分, 其余 3 分)

输入: 三个不超过 10000 的正整数 x , p , m 。

输出: $x^p \bmod m$ 的值。

提示: 若 p 为偶数, $x^p = (x^2)^{p/2}$; 若 p 为奇数, $x^p = x * (x^2)^{(p-1)/2}$ 。

```

#include <iostream>
using namespace std;

```

```

int x, p, m, i, result;

```

```

int main() {
    cin >> x >> p >> m;
    result = __ (1) __;
    while ( __ (2) __ ) {
        if (p % 2 == 1)
            result = __ (3) __;
        p /= 2;
        x = __ (4) __;
    }
    cout << __ (5) __ << endl;
    return 0;
}

```

答案 (1)1
(2) $p > 0$ 或 $p \neq 0$ 或 p
(3) $\text{result} * x \% m$
(4) $x * x \% m$
(5)result

第 28 题 问答题

(切割绳子) 有 n 条绳子，每条绳子的长度已知且均为正整数。绳子可以以任意正整数长度切割，但不能连接。现在要从这些绳子中切割出 m 条长度相同的绳段，求绳段的最大长度是多少。（第一、二空 2.5 分，其余 3 分）

输入：第一行是一个不超过 100 的正整数 n ，第二行是 n 个不超过 106 的正整数，表示每条绳子的长度，第三行是一个不超过 108 的正整数 m 。

输出：绳段的最大长度，若无法切割，输出 Failed。

```
#include <iostream>
using namespace std;

int n, m, i, lbound, ubound, mid, count;
int len[100]; // 绳子长度
int main() {
    cin >> n;
    count = 0;
    for (i = 0; i < n; i++) {
        cin >> len[i];
        _____(1)_____;
    }
    cin >> m;
    if (____(2)____) {
        cout << "Failed" << endl;
        return 0;
    }
    lbound = 1;
    ubound = 1000000;
    while (____(3)____) {
        mid = ____ (4) ____;
        count = 0;
        for (i = 0; i < n; i++)
            _____(5)_____;
        if (count < m)
```

```

        ubound = mid - 1;
    else
        lbound = mid;
    }
    cout << lbound << endl;
    return 0;
}

```

答案

```

(1)count=count+len[i] 或 count+=len[i]
(2)count<m 或 m>count
(3)lbound<ubound 或 ubound>lbound
(4)(lbound+ubound+1)/2 或 (lbound+ubound+1)>>1 或 (lbound+ubound)/2+1
(5)count=count+len[i]/mid 或 count+=len[i]/mid

```