

## 第二十届全国青少年信息学奥林匹克联赛初赛

### 提高组 C++语言试题

竞赛时间：2014 年 10 月 12 日 14:30~16:30

#### 选手注意：

- 试题纸共有 10 页，答题纸共有 2 页，满分 100 分。请在答题纸上作答，写在试题纸上的一律无效。
- 不得使用任何电子设备（如计算器、手机、电子词典等）或查阅任何书籍资料。

#### 一、单项选择题（共 15 题，每题 1.5 分，共计 22.5 分；每题有且仅有一个正确选项）

1. 以下哪个是面向对象的高级语言（ ）。

- A. 汇编语言      B. C++      C. Fortran      D. Basic

计算机语言具有高级语言和低级语言之分。而高级语言又主要是相对于汇编语言而言的高级语言并不是特指的某一种具体的语言，而是包括很多编程语言，如流行的 java，c，c++，C#，pascal，python，lisp，prolog，FoxPro，易语言

低级语言分机器语言（二进制语言）和汇编语言（符号语言），这两种语言都是面向机器的语言，和具体机器的指令系统密切相关。机器语言用指令代码编写程序，而符号语言用指令助记符来编写程序。

2. 1TB 代表的字节数量是（ ）。

- A. 2 的 10 次方      B. 2 的 20 次方      C. 2 的 30 次方      D. 2 的 40 次方

D

$1TB=2^{10}GB=2^{20}MB=2^{30}KB=2^{40}BYTE$

3. 二进制数 00100100 和 00010101 的和是（ ）。

- A. 00101000      B. 001010100      C. 01000101      D. 00111001

送分题，了解二进制计算即可。

4. TCP 协议属于哪一层协议（ ）。A. 应用层 B. 传输层 C. 网络层 D. 数据链路层

B

TCP（传输控制协议）

5. 下列几个32位IP地址中，书写错误的是（ ）。A. 162.105.136.27 B. 192.168.0.1 C. 256.256.129.1 D. 10.0.0.1

C

IP地址0~255

6. 在无向图中，所有顶点的度数之和是边数的（ ）倍。A. 0.5 B. 1 C. 2 D. 4

C

随便画个图看一下就行了，原理是因为无向图中，每个点的入度和=边数，每条边算了2遍

7. 对长度为n的有序单链表，若检索每个元素的概率相等，则顺序检索到表中任一元素的平均检索长度为（ ）。A.  $n/2$  B.  $(n+1)/2$  C.  $(n-1)/2$  D.  $n/4$

B

检索任一元素检索长度 $T=1+2+3+4+5+6+\dots+n=n(n+1)/2$

因为概率相等，所以结果为 $T/n=(n+1)/2$

8. 编译器的主要功能是（ ）。

A. 将一种高级语言翻译成另一种高级语言 B. 将源程序翻译成指令

C. 将低级语言翻译成高级语言 D. 将源程序重新组合

B

9. 二进制数111.101所对应的十进制数是（ ）。A. 5.625 B. 5.5

C. 6.125

D. 7.625

D

10. 若有变量 `int a, float x, y`, 且  $a=7, x=2.5, y=4.7$ , 则表达式  $x+a\%3*(int)(x+y)\%2/4$  的值大约是（ ）。

A. 2.500000

B. 2.750000

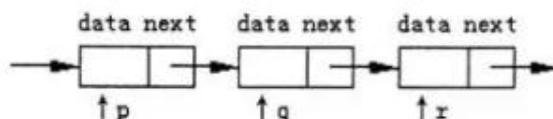
C. 3.500000

D. 0.000000

A 运算优先级问题: 同一行的同级别，按从左往右依次计算

11. 有以下结构体说明和变量定义，如图所示，指针 `p`、`q`、`r` 分别指向一个链表中的三个连续结点。

```
struct node {
    int data;
    node *next;
} *p, *q, *r;
```



现要将  $q$  和  $r$  所指结点的先后位置交换，同时要保持链表的连续，以下程序段中错误的是（ ）。

- A.  $q \rightarrow next = r \rightarrow next; p \rightarrow next = r; r \rightarrow next = q;$
- B.  $p \rightarrow next = r; q \rightarrow next = r \rightarrow next; r \rightarrow next = q;$
- C.  $q \rightarrow next = r \rightarrow next; r \rightarrow next = q; p \rightarrow next = r;$
- D.  $r \rightarrow next = q; q \rightarrow next = r \rightarrow next; p \rightarrow next = r;$

D

只要学过链表水过就好

12. 同时查找  $2n$  个数中的最大值和最小值，最少比较次数为（ ）。

- A.  $3(n-2)/2$
- B.  $4n-2$
- C.  $3n-2$
- D.  $2n-2$

C

其实我是蒙不对的，于是题解大人教导到：

前两个数比较，大的为最大值，小的为最小值，用掉1次；

还剩下  $2 * (n-1)$  个数，每两个比较，大的再和最大值比较，小的再和最小值比较，一共是  $3 * (n-1)$  次所以加一起就是  $(3 * n - 2)$  次13. 设  $G$  是有6个结点的完全图，要得到一棵生成树，需要从  $G$  中删去（ ）条边。A. 6 B. 9

- C. 10
- D. 15

C

一个图的每一对不同顶点恰有一条边相连，则称为完全图

一个完全图共有  $n * (n-1) / 2$  条边，树有  $(n-1)$  条边

So，答案就出来了

其实随便画一下再数一下也就出来了=w=

14. 以下时间复杂度不是 $O(n^2)$ 的排序方法是 ( )。

- A. 插入排序
- B. 归并排序
- C. 冒泡排序
- D. 选择排序

**B, 显然**

由于出现频率较高, 所以各种排序的时间复杂度大家一定要牢记!

15. 以下程序段实现了找第二小元素的算法。输入是  $n$  个不等的数构成的数组  $S$ , 输出  $S$  中第二小的数  $SecondMin$ 。在最坏情况下, 该算法需要做 ( ) 次比较。

```

if (S[1] < S[2]) {
    FirstMin = S[1];
    SecondMin = S[2];
} else {
    FirstMin = S[2];
    SecondMin = S[1];
}
for (i = 3; i <= n; i++)
    if (S[i] < SecondMin)
        if (S[i] < FirstMin) {
            SecondMin = FirstMin;
            FirstMin = S[i];
        } else {
            SecondMin = S[i];
        }
    
```

- A.  $2n$
- B.  $n-1$
- C.  $2n-3$
- D.  $2n-2$

C 前两个数比较用掉1次, 剩下  $(n-2)$ 个数 最坏的情况剩下每个数比较两次 所以再加起来, 得C

1. 若逻辑变量  $A$ 、 $C$  为真,  $B$ 、 $D$  为假, 以下逻辑运算表达式为真的有 ( )。

- A.  $(B \vee C \vee D) \vee D \wedge A$
- B.  $((\neg A \wedge B) \vee C) \wedge \neg B$
- C.  $(A \wedge B) \vee (C \wedge D \vee \neg A)$
- D.  $A \wedge (D \vee \neg C) \wedge B$

AB 逻辑运算，不明白每个符号表示意义就是基础不够。

2. 下列 ( ) 软件属于操作系统软件。A. Microsoft Word B. Windows XP C. Android D. Mac OS X E. Oracle

BCD

其实BC应该是都能蒙出来的，A显然是不选的（文字编译软件），至于DE姿势不够的话就只能靠人品了

典型的操作系统：Linux、Windows (XP)、Android、Mac OS X、IOS、WP、Chrome OS

3. 在NOI比赛中，对于程序设计题，选手提交的答案不得包含下列哪些内容 ( )。A. 试图访问网络

B. 打开或创建题目规定的输入/输出文件之外的其他文件 C. 运行其他程序

D. 改变文件系统的访问权限 E. 读写文件系统的管理信息

ABCDE

这都是套路=。=

4. 以下哪些结构可以用来存储图 ( )。A. 邻接矩阵 B. 栈

C. 邻接表

D. 二叉树

AC

学过图的应该都没问题=w=

5. 下列各无符号十进制整数中，能用八位二进制表示的数有 ( )。

A. 296

B. 133

C. 256

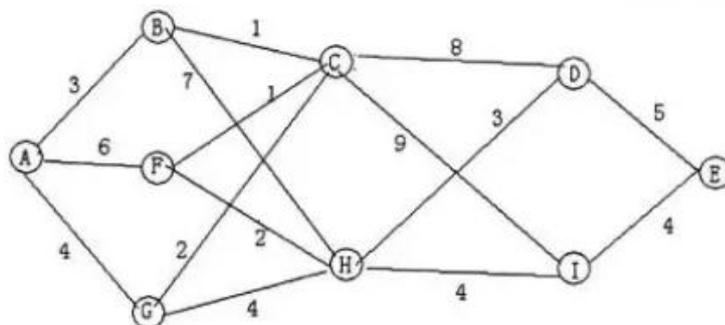
D. 199

BD

八位二进制表示范围 $[2^7, 2^8)$ （左闭右开）

三、问题求解（共2题，每题5分，共计10分；每题全部答对得5分，没有部分分）

1. 由数字 1, 1, 2, 4, 8, 8 所组成的不同的四位数的个数是\_\_\_\_\_。
2. 如图所示, 图中每条边上的数字表示该边的长度, 则从 A 到 E 的最短距离是\_\_\_\_\_。



- 1.102 排列组合去算或者把所有情况列出来=。= (不要上来就选后者, 考虑过脑子的感受么  
=A=, 你总也不用它合适么)
- 2.20 学过最短路的应该都能出答案而且还能检验着玩

#### 四、阅读程序写结果 (共 4 题, 每题 8 分, 共计 32 分)

```

1. #include <iostream>
   using namespace std;

   int main() {
       int a, b, i, tot, c1, c2;
       cin >> a >> b;
       tot = 0;
       for (i = a; i <= b; i++)
       {
           c1 = i / 10;
           c2 = i % 10;

           if ((c1 + c2) % 3 == 0)
               tot++;
       }
       cout << tot << endl;
       return 0;
   }

```

CCF NOIP2014 初赛提高组 C++ 语言试题  
第 4 页, 共 10 页

}

输入: 7 31

输出: \_\_\_\_\_

答案: 8 (就是在[a,b]范围内找3的倍数)

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

int fun(int n, int minNum, int maxNum) {
    int tot, i;
    if (n == 0)
        return 1;
    tot = 0;
    for (i = minNum; i <= maxNum; i++)
        tot += fun(n - 1, i + 1, maxNum);
    return tot;
}

int main() {
    int n, m;
    cin >> n >> m;
    cout << fun(m, 1, n) << endl;
    return 0;
}
```

输入: 6 3

输出: \_\_\_\_\_

答案: 20 (写一写, 导一导, 递归)

$Fun(3,1,6)=fun(2,2,6)+fun(2,3,6)+fun(2,4,6)+fun(2,5,6)+fun(2,6,6)+fun(2,7,6)=20$

$Fun(2,2,6)=fun(1,3,6)+fun(1,4,6)+fun(1,5,6)+fun(1,6,6)+f(1,7,6)=10$

$Fun(2,3,6)=fun(1,4,6)+fun(1,5,6)+fun(1,6,6)+fun(1,7,6)=6$

$Fun(2,4,6)=fun(1,5,6)+fun(1,6,6)+fun(1,7,6)=3$

$Fun(2,5,6)=fun(1,6,6)+fun(1,7,6)=1$

Fun(2,6,6):=fun(1,7,6)=0

Fun(1,3,6)=fun(0,4,6)+fun(0,5,6)+fun(0,6,6)+fun(0,7,6)=4

Fun(1,4,6)=fun(0,5,6)+fun(0,6,6)+fun(0,7,6)=3

Fun(1,5,6)=fun(0,6,6)+fun(0,6,7)=2

Fun(1,6,6)=fun(0,7,6)=1

Fun(1,7,6)=0

```

3. #include <iostream>
   #include <string>
   using namespace std;

const int SIZE = 100;

int main() {
    string dict[SIZE];
    int rank[SIZE];
    int ind[SIZE];
    int i, j, n, tmp;
    cin >> n;
    for (i = 1; i <= n; i++) {
        rank[i] = i;
        ind[i] = i;
        cin >> dict[i];
    }
    for (i = 1; i < n; i++)
        for (j = 1; j <= n - i; j++)
            if (dict[ind[j]] > dict[ind[j + 1]]){
                tmp = ind[j];
                ind[j] = ind[j + 1];
                ind[j + 1] = tmp;
            }
    for (i = 1; i <= n; i++)

```

```
        rank[ind[i]] = i;
    for (i = 1; i <= n; i++)
        cout << rank[i] << " ";
    cout << endl;
    return 0;
}
```

输入:

```
7
aaa
aba
bbb
aaa
aaa
ccc
aa
```

输出: \_\_\_\_\_

答案: 2 5 6 3 4 7 1 (看到那个冒泡排序就应该差不多了吧,就是输出各个字符按字典序排序后应该在的位置)

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

    const int SIZE = 100;

    int alive[SIZE];
    int n;

    int next(int num) {
        do {
            num++;
            if (num > n)
                num = 1;
        }
```

```

        } while (alive[num] == 0);
        return num;
    }

int main() {
    int m, i, j, num;
    cin >> n >> m;
    for (i = 1; i <= n; i++)
        alive[i] = 1;
    num = 1;
    for (i = 1; i <= n; i++) {
        for (j = 1; j < m; j++)
            num = next(num);
        cout << num << " ";
        alive[num] = 0;
        if (i < n)
            num = next(num);
    }
    cout << endl;
    return 0;
}

```

输入: 11 3

输出: \_\_\_\_\_

答案: 3 6 9 1 5 10 4 11 8 2 7 (约瑟夫环问题, 用链表实现, 11个人围一圈, 从1开始报数, 报到3的出局, 再从出局的下一个人开始报1, 直到全部出局, 依次输出出局人的编号)

### 五、完善程序 (每题 14 分, 共计 28 分)

1. (双栈模拟数组) 只使用两个栈结构 `stack1` 和 `stack2`, 模拟对数组的随机读取。作为栈结构, `stack1` 和 `stack2` 只能访问栈顶 (最后一个有效元素)。栈顶指针 `top1` 和 `top2` 均指向栈顶元素的下一个位置。

输入第一行包含两个整数, 分别是数组长度 `n` 和访问次数 `m`, 中间用单个空格隔开。第二行包含 `n` 个整数, 依次给出数组各项 (数组下标从 0 到 `n-1`)。第三行包含 `m` 个整数, 需要访问的数组下标。对于每次访问, 输出对应的数组元素。(前两空每空 2.5 分,

其余每空 3 分，共 14 分)

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

const int SIZE = 100;

int stack1[SIZE], stack2[SIZE];
int top1, top2;
int n, m, i, j;
void clearStack() {
    int i;
    for (i = top1; i < SIZE; i++)
        stack1[i] = 0;
    for (i = top2; i < SIZE; i++)
        stack2[i] = 0;
}

int main() {
    cin >> n >> m;
    for (i = 0; i < n; i++)
        cin >> stack1[i];
    top1 = (1);
    top2 = (2);
    for (j = 0; j < m; j++) {
        cin >> i;
        while (i < top1 - 1) {
            top1--;
            (3);
            top2++;
        }
        while (i > top1 - 1) {
            top2--;
            (4);
        }
    }
}
```

```

        top1++;
    }
    clearStack();
    cout << stack1[ (5) ] << endl;
}
return 0;
}

```

2. (最大子矩阵和) 给出  $m$  行  $n$  列的整数矩阵, 求最大的子矩阵和 (子矩阵不能为空)。

输入第一行包含两个整数  $m$  和  $n$ , 即矩阵的行数和列数。之后  $m$  行, 每行  $n$  个整数, 描述整个矩阵。程序最终输出最大的子矩阵和。(第一空 2 分, 其余 3 分, 共 14 分)

```

#include <iostream>
using namespace std;

const int SIZE = 100;

int matrix[SIZE + 1][SIZE + 1];
int rowsum[SIZE + 1][SIZE + 1]; //rowsum[i][j]记录第 i 行前 j 个数的和
int m, n, i, j, first, last, area, ans;

int main() {
    cin >> m >> n;
    for (i = 1; i <= m; i++)
        for (j = 1; j <= n; j++)
            cin >> matrix[i][j];
    for (i = 1; i <= m; i++)
        (2);
    for (i = 1; i <= m; i++)
        for (j = 1; j <= n; j++)
            rowsum[i][j] = (3);
    for (first = 1; first <= n; first++)
        for (last = first; last <= n; last++) {
            (4);
            for (i = 1; i <= m; i++) {

```

```

        area += (5);
        if (area > ans)
            ans = area;
        if (area < 0)
            area = 0;
    }
}
cout << ans << endl;
return 0;
}

```

五、完善程序（共计 28 分，以下各程序填空可能还有一些等价的写法，由各省赛区组织本省专家审定及上机验证，可以不上报 CCF NOI 科学委员会复核）

		Pascal 语言	C++语言	C 语言	分值
1.	(1)		n		2.5
	(2)		0		2.5
	(3)	stack2[top2]:=stack1[top1]	stack2[top2]=stack1[top1]		3
	(4)	stack1[top1]:=stack2[top2]	stack1[top1]=stack2[top2]		3
	(5)		top1-1		3
2.	(1)	[1,1]	[1][1]		2
	(2)	rowsum[i,0]:=0	rowsum[i][0]=0		3
	(3)	rowsum[i,j-1]+matrix[i][j]	rowsum[i][j-1]+matrix[i][j]		3
	(4)	area:=0	area=0		3
	(5)	rowsum[i,last]-rowsum[i,first-1]	rowsum[i][last]-rowsum[i,first-1]		3

北京高考在线是长期为中学老师、家长和考生提供新鲜的高考资讯、专业的高考政策解读、科学的升学规划以及实用的升学讲座活动等全方位服务的升学服务平台。自 2014 年成立以来一直致力于服务北京考生，助力千万学子，圆梦高考。

目前，北京高考在线拥有旗下拥有北京高考在线网站和北京高考资讯微信公众号两大媒体矩阵，关注用户超 20 万+。

北京高考在线\_2020 年北京高考门户网站

<http://www.gaokzx.com/>

北京高考资讯微信：bj-gaokao

## 北京高考资讯

### 关于我们

北京高考资讯隶属于太星网络旗下，北京地区高考领域极具影响力的升学服务平台。

北京高考资讯团队一直致力于提供最专业、最权威、最及时、最全面的高考政策和资讯。期待与更多中学达成更广泛的合作和联系。

长按二维码 识别关注



微信公众号：bj-gaokao

官方网址：www.gaokzx.com

咨询热线：010-5751 5980