第?届全国青少年信息学奥林匹克联赛初赛

普及组 C++ 语言试题

竞赛时间：2020年10月?日 ??:??~??:??

**选手注意：**不得使用任何电子设备（手机、计算器），或查阅任何书籍资料。满分100分。

**一、单项选择题（共20题，每题1.5分，共计30分）**

**1、使用高级语言编写的程序称之为：（）**

A、应用程序 B、源程序

C、运行程序 D、高级程序

**2、以下哪一对快捷键与功能的组合是正**确的？（）

A、Ctrl+A 截屏 B、Shift+Space 切换输入法

C、Alt+F4 关机 D、Ctrl+Y 恢复

**3、调制解调器Modem（俗称“猫”）属于哪一种设备？**

A、输入设备 B、输出设备

C、内部设备 D、外部设备

**4、www.google.com是谷歌的域名，其中www是哪一级域名？**

A、三级域名 B、二级域名

C、一级域名 D、顶级域名

**5、以下哪一款软件属于IDE（集成开发环境）？**

A、Wireshark B、Multisim

C、WebStorm D、Davinci

**6、以下哪一个厂商没有推出过手机产品？**

A、Google B、Microsoft

C、Apple D、Facebook

**7、Linux系统下具有最高权限的用户是？**

A、root B、admin

C、Administrator D、User

**8、以下哪一个编码方式不是中文编码？**

A、Unicode B、UTF-8

C、ASCII D、GB2312

**9、与十进制数17.5625 对应的8 进制数是？**

A、21.5625 B、21.44

C、21.73 D、以上都不是

**10、二进制小数1.010101...（01无限循环）对应的十进制数是？**

A、4/3 B、5/4

C、7/6 D、3/2

**11、堆排序是一种（）排序？**

A、插入 B、选择

C、交换 D、归并

**12、下列代码的时间复杂度是？**

x=n,y=1;

while(x\*x>=(y+1)\*(y+1))

y+=y**;**

A、 B、

C、 D、

**13、关于计算机网络，下列说法错误的是？**

A、网络协议之所以有很多层主要是由于新技术需要兼容过去老的实现方案。

B、新一代互联网使用的IPv6标准是IPv5标准的升级与补充。

C、TCP/IP是互联网基础协议簇，包含有TCP和IP等网络与传输层的通讯协议。

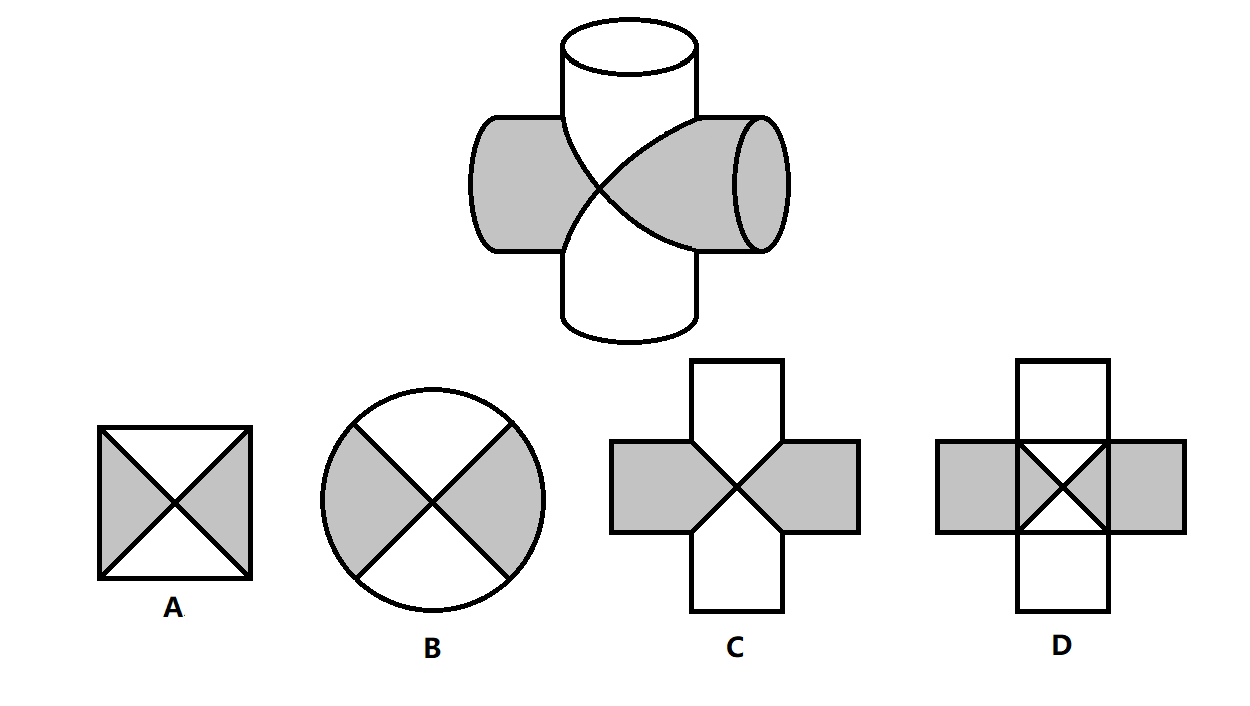
D、互联网上每一台入网主机通常都需要使用一个唯一的IP地址，否则就必须注册一个固定的域名来标明其地址。

**14、在字长为16位的系统环境下，一个16位带符号整数的二进制补码为1111111111101101。其对应的十进制整数应该是？**

A、18 B、-18

C、19 D、-19

**15、牟合方盖指的是两个相同的圆柱互相垂直交叉时的重合部分。如图。牟合方盖从正面看过去，应该是什么样的？（正视图）**



**16、一个正整数在十六进制下有100位，则它在二进制下可能有（ ）位**

A、396 B、399

C、401 D、404

**17、G 是一个非连通简单无向图，共有 28 条边，则该图至少有（ ）个顶点。**

A、10  **B、9**

C、8 D、7

**18、从4个班级中选7名同学参赛，不要求每个班必须选，有（ ）种不同的选取方案。**

A、60 B、84

C、96 D、120

**19、若某算法的计算时间表示为递推关系式：T(N)=2T(N/2)+NlogN，T(1)=1**

**则该算法的时间复杂度为（ ）。**

A、O(N) B、O(NlogN)

C、O(N log^2 N) D、O(N^2）

**20、下列哪种行为是不被严格禁止的？**

A、在考试过程中向同学咨询题目相关问题。

B、在程序中启动多个进程以提高程序的运行效率。

C、使用计算器直接算出题目答案写在答题纸上。

D、考试时在网盘里寻找过往保存的OI资料

**二、问题求解（共2题，每题5分，共10分）**

**21、一棵有5个节点的树，共有 种不同的形态。**

**Prufer序列**

**22、七人出游拍照站成一排，其中A与B不相邻，共有 种不同的站法。**

**A(7,7)-A(6,6)\*A(2,2) = 4800**

**三、读程序写结果（共4题，每题8分，共32分）**

**1.**

#include<cstdio>

int main(){

int i, p[5], a, b, c, x, y = 20;

for ( i = 0; i <= 4; i++ ) scanf( "%d", &p[i] );

a = (p[0] + p[1]) + (p[2] + p[3] + p[4]) / 7;

b = p[0] + p[1] / ( (p[2] + p[3]) / p[4]);

c = p[0] \* p[1] / p[2];

x = a + b - p[(p[3] + 3) % 4];

if (x>10) y += (b \* 100 - a) / (p[p[4] % 3] \* 5);

else y += 20 + (b \* 100 - c) / (p[p[4] % 3] \* 5);

printf( "%d,%d\n", x, y );

return 0;

}

输入：6 6 5 5 3

输出：

2.

#include<iostream>

using namespace std;

int main(){

const int SIZE=10;

int data[SIZE],i,j,cnt,n,m;

cin>>n>>m;

for(i=1;i<=n;i++) cin>>data[i];

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

cnt=0;

for(j=1;j<=n;j++)

if((data[i]<data[j])||(data[j]==data[i]&&j<i))

cnt++;

if (cnt==m)

cout<<data[i]<<endl;

}

return 0;

}

输入：

5 2  
96 -8 0 16 87

输出：

**3．**

#include <cstdio>

char ch[] = { 'q', 'A', 'S', 'O', 'R', 'T', 'E', 'X', 'A', 'M', 'P', 'L', 'E' };

int n = 12;

void shift( int k, int n ){

char v;

int j;

v = ch[k]; j = k + k;

while ( j <= n ){

if ( (j < n) && (ch[j] < ch[j + 1]) ) j++;

if ( v < ch[j] ){ ch[j / 2] = ch[j]; j \*= 2;}

else return;

ch[j / 2] = v;

}

}

void hpsrt(){

int k;

char tmp;

for ( k = n / 2; k > 0; k-- ) shift( k, n );

printf( "No.1: " );

for ( k = 1; k <= n; k++ ) putchar( ch[k] );

putchar( '\n' );

for ( k = n; k > 0; k-- ) {

tmp = ch[1]; ch[1] = ch[k]; ch[k] = tmp;

shift( 1, k - 1 );

}

}

int main(){

int k;

hpsrt();

printf( "No.2: " );

for ( k = 1; k <= n; k++ ) putchar( ch[k] );

putchar( '\n' );

}

输出：

4.

#include<iostream>

using namespace std;

const int NUM=5;

int r(int n){

int i;

if(n<=NUM) return 0;

for(i=1;i<=NUM;i++)

if( r(n-i)<0)

return i;

return -1;

}

int main(){

int n;

cin>>n;

cout<<r(n)<<endl;

return 0;

}

输入：16

输出：

**四、完善程序（共2题，每题14分，共28分）**

1.(最大矩阵和)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 <stdio.h>

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(){

scanf(“%d %d”,&m,&n);

for(i=1;i<=m;i++)

for(j=1;j<=n;j++)

scanf(“%d”,&matrix[i][j]);

ans=matrix ;

for(i=1;i<=m;i++)

for(i=1;i<=m;i++)

for(j=1;j<=n;j++)

rowsum[i][j]= ;

for(first=1;first<=n;first++)

for(last=first;last<=n;last++){

;

for(i=1;i<=m;i++){

area+= ;

if(area>ans) ans=area;

if(area<0) area=0;

}

}

printf(“%d\n”,ans);

return 0;

}

2. （最短路径问题）无向连通图 G 有 n 个结点，依次编号为 0,1,2,...,(n-1)。用邻接矩阵的 形式给出每条边的边长，要求输出以结点 0 为起点出发，到各结点的最短路径长度。 使用 Dijkstra 算法解决该问题：利用 dist 数组记录当前各结点与起点的已找到的最 短路径长度；每次从未扩展的结点中选取 dist 值最小的结点 v 进行扩展，更新与 v 相邻 的结点的 dist 值；不断进行上述操作直至所有结点均被扩展，此时 dist 数据中记录的值 即为各结点与起点的最短路径长度。（第五空 2 分，其余 3 分）

#include <iostream>

using namespace std;

const int MAXV = 100;

int n, i, j, v; int w[MAXV][MAXV]; // 邻接矩阵，记录边长

// 其中 w[i][j]为连接结点 i 和结点 j 的无向边长度，若无边则为-1

int dist[MAXV]; int used[MAXV];

// 记录结点是否已扩展（0：未扩展；1：已扩展）

int main() {

cin >> n;

for (i = 0; i < n; i++)

for (j = 0; j < n; j++)

cin >> w[i][j];

dist[0] = 0;

for (i = 1; i < n; i++) dist[i] = -1;

for (i = 0; i < n; i++) used[i] = 0;

while (true) {

for (i = 0; i < n; i++)

if(used[i]!=1&&dist[i]!=-1&&(v==-1|| ))

;

if (v == -1) break;

;

for (i = 0; i < n; i++)

if (w[v][i]!=-1&&(dist[i]==-1|| ))

dist[i]=dist[v]+w[v][i];

}

for (i = 0; i < n; i++) cout << dist[i] << endl;

return 0;

}