

2021 CCF 非专业级别软件能力认证第一轮

(CSP-J)入门级 C++语言试题 模拟卷 - 8

考生注意事项:

1. 全部试题答案均要求写在答卷纸上, 写在试卷纸上一律无效。
2. 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、单选题 (共 15 题, 每题 2 分, 共计 30 分; 每题有且仅有一个正确答案)

1. $2E+03$ 表示 ()。

- A. 2.03 B. 5 C. 8 D. 2000

2. 一个字节 (byte)由 () 个二进制位组成。

- A. 8 B. 16 C. 32 D. 以上皆有可能

3. 以下逻辑表达式的值恒为真的是 ()。

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

4. 在一个无向的简单图 G 中, 共有 30 条边, 问至少有多少个顶点。无向图中的简单图指的是无重边, 无自环。

- A. 30 B. 31 C. 9 D. 8

5. 提出“存储程序”的计算机原理的是 ()。

- A. 克劳德·香农 B. 戈登·摩尔
C. 查尔斯·巴比奇 D. 冯·诺依曼

6. 班里有 7 位同学，每位同学都准备了一份礼物，老师负责分配礼物给同学，并且同学不会拿到自己准备的礼物，问老师有多少种分配礼物的方案。

- A. 265 B. 768 C. 1854 D. 14833

7. 前缀表达式 “+ 3 * 2 + 5 12” 的值是 () 。

- A.23 B.25 C.37 D.65

8. 主存储器的存取速度比中央处理器 (CPU)的工作速度慢得多，从而使得后者的效率受到影响。而根据局部性原理，CPU 所访问的存储单元通常都趋于聚集在一个较小的连续区域中。于是，为了提高系统的整体执行效率，在 CPU 中引入 () 。

- A.寄存器 B.高速缓存 C.闪存 D.外存

9. 对 8 位的十进制数 3 和 8，执行(3 << 8 >> 8)的返回值是 () 。

- A. 3 B. 0
C. 1 D. 6

10. 班里有 7 个同学参与排队，A 想和 B 排在一起，但是 A 不想和 C 排在一起，问有多少种排队的方案 () 。

- A. 1200 B. 1440 C. 600 D. 720

11. 一个自然数在十进制下有 n 位，则它在二进制下的位数与 () 最接近。

- A.5n B. $n\log_2 10$ C. $10\log_2 n$ D. $10n\log_2 n$

12. 元素 R1、R2、R3、R4、R5 入栈的顺序为 R1、R2、R3、R4、R5.如果第一个出栈的是 R3,那么第五个出栈的不可能是 () 。

- A.R1 B.R2 C.R4 D.R5

13.双向链表中有两个指针域 llink 和 rlink,分别指向该结点的前驱和后继。设 P 指向链表中的一个结点, 它的左右结点均非空。现要求删除结点 P,则下面语句序列中错误的是 () 。

- A.P->rlink->llink=P->rlink;P->llink->rlink=P->llink; delete(P)
B.P->llink->rlink=P->rlink;P->rlink->llink=P->llink; delete(P)
C.P->rlink->llink=P->llink;P->rlink->llink->rlink=P->rlink;delete(P)
D.P->llink->rlink=P->rlink;P->llink->rlink->llink=P->llink;delete(P)

14.一棵二叉树的前序遍历序列是 ABCDEFG,后序遍历序列是 CBFEGDA,则根结点的左子树的结点个数可能是 () 。

- A.2 B.3 C.4 D.5

15.完全二叉树的顺序存储方案,是指将完全二叉树的结点从上至下、从左至右依次存放到一个顺序结构的数组中。假定根结点存放在数组的 1 号位置,则第 k 号结点的父结点如果存在的话,应当存放在数组的 () 号位置。

- A.2k B.2k+1
C.k/2 向下取整 D.(k+1)/2 向下取整

二、阅读程序 (程序输入不超过数组或字符串定义的范围: 判断题正确填“√”, 错误填“×”; 除特殊说明外, 判断题 1.5 分, 选择题 4 分, 共计 40 分)

01	#include<iostream>
0	using namespace std;
2	
0	int rSum(int j)
3	{
0	int sum=0;
4	while(j!=0)
0	{
5	sum=sum*10+(j%10);
0	j=j/10;
6	}
07	return sum;
0	}
8	
0	int main()
9	{
10	int n,m,i;
11	cin>>n>>m;
12	for(i=n;i<m;i++)
13	if(i==rSum(i))
14	cout<<i<<' ';

2.

01	#include<iostream>
0	#include<string>
2	using namespace std;
0	
3	int main()
0	{
4	string s;
0	char m1,m2;
5	int i;
0	getline(cin,s);
6	m1=' ';
07	m2=' ';
0	for(i=0;i<s.length();i++)
8	if(s[i]>m1)
0	{
9	m2=m1;
10	m1=s[i];
11	}
12	else if(s[i]>m2)
13	m2=s[i];

14	cout<<int(m1)<<' '<<int(m2)<<endl;
15	return 0;
16	}
17	
18	
19	
2	
0	
21	
2	
2	
2	
3	

字符	空格	'0'	'A'	'a'
ASCII 码	32	48	65	97

判断题

- 1)输入的字符串中包含有空格时，程序不能完整读入()
- 2)若输入的字符串只包含小写字母，输出的两个值一定大于等于 97。()
- 3)若输入的字符串只包含大小写字母，且输出的第二个值为 90，则输入的字符串中有且只有一个小写字母。()
- 4)若输入的字符串只包含数字 2，且多于两个字符，则输出的两个值都是 50。()

选择题

5)若输入的字符串为“Expo 2010 Shanghai China”，则输出的结果是()。

A.120 110

B.120 112

C.110 120

D.112 120

6)若输入的字符串有 3 个字符，且都是数字，并且输出的两个值分别是 51 和 50，则输入一共有()种不同的方案。

A.7

B.15

C.31

D.63

3.

```
01  #include<iostream>
0   using namespace std;
2   const int NUM=5;
0
3   int r(int n)
0   {
4       int i;
0       if(n<=NUM)
5           return n;
0       for(i=1;i<=NUM;i++)
6           if(r(n-i)<0)
07              return i;
0       return -1;
8   }
```


0	
9	int main()
10	{
11	int n;
12	cin>>n;
13	cout<<r(n)<<endl;
14	return 0;
15	}
16	
17	
18	
19	
2	
0	
21	
2	
2	

保证输入为非负整数数字。

判断题

- 1)将第 10 行 “i=1” 改为 “i=0”, 程序不会出错。()
- 2)程序输出的结果有可能小于-1。()
- 3)若程序两次输入的值分别为 n_1 和 n_2 , 且有 $n_1 - n_2 = 1$ 的关系, 则对于这两次运行的结果 ans_1 和 ans_2 , 有 $ans_1 - ans_2 = 1$ 。()

4)若输入的 n 大于等于 6 时, 程序一定至少执行一次第 13 行。()

选择题

5)(2 分)若输入 2020, 输出的结果为 ()。

A.3 B.4 C.5 D.-1

6)若已知 $0 \leq n \leq 100$, 则要使输出的结果为 -1, 则 n 的取值有()种。

A.10 B.12 C.14 D.16

三、完善程序(单选题,每小题 3 分,共计 30 分)

1.(哥德巴赫猜想)哥德巴赫猜想是指, 任一大于 2 的偶数都可写成两个质数之和。迄今为止, 这仍然是一个著名的世界难题, 被誉为数学王冠上的明珠。试编写程序, 验证任一大于 2 且不超过 n 的偶数都能写成两个质数之和。

01	#include<iostream>
0	using namespace std;
2	int main()
0	{
3	const int SIZE=1000;
0	int n,r,p[SIZE],i,j,k,ans;
4	bool tmp;
0	cin>>n;
5	r=1;

0	p[1]=2;
6	for(i=3;i<=n;i++)
07	{
0	①;
8	for(j=1;j<=r;j++)
0	if(i%②==0)
9	{
10	tmp=false;
11	break;
12	}
13	if(tmp)
14	{
15	r++;
16	③;
17	}
18	}
19	ans=0;
2	for(i=2;i<=n/2;i++)
0	{
21	tmp=false;
2	for(j=1;j<=r;j++)
2	for(k=j;k<=r;k++)

2	if(i+i==④)
3	{
2	tmp=true;
4	break;
2	}
5	if(tmp)
2	ans++;
6	}
27	cout<<ans<<endl;
2	return 0;
8	}
2	
9	
3	
0	
31	
3	
2	
3	
3	
3	
4	

3	
5	
3	
6	
37	
3	
8	
3	
9	
4	
0	
41	
4	
2	

若输入 n 为 2010，则输出 ⑤ 时表示验证成功，即大于 2 且不超过 2010 的偶数都满足哥德巴赫猜想。

1)①处应填()。

A. tmp=true

B. tmp=false

C. tmp++

D. r=sqrt(n)

2)②处应填()。

A. j

B. p[j]

C. p[j+1]

D. p[i]

3)③处应填()。

- A. $p[r]=i$ B. $p[r-1]=i$ C. $p[r]=j$ D. $p[r-1]=j$

4)④处应填()。

- A. $p[k]$ B. $p[i]+p[k]$ C. $p[i]+p[i]$ D. $p[j]+p[k]$

5)⑤处应填()。

- A.1003 B. 1004 C.1005 D.1006

2.(过河问题)在一个月黑风高的夜晚，有一群人在河的右岸，想通过唯一的一根独木桥走到河的左岸。在伸手不见五指的黑夜里，过桥时必须借照灯光来照明，不幸的是，他们只有一盏灯。另外,独木桥上最多能承受两个人同时经过，否则将会坍塌。每个人单独过独木桥都需要一定的时间，不同的人用的时间可能不同。两个人一起过独木桥时,由于只有一盏灯，所以需要的时间是较慢的那个人单独过桥所花费的时间。现在输入 $N(2 \leq N < 1000)$ 和这 N 个人单独过桥需要的时间，请计算总共最少需要多少时间，他们才能全部到达河左岸。

例如，有 3 个人甲、乙、丙，他们单独过桥的时间分别为 1、2、4,则总共最少需要的时间为 7。具体方法是:甲、乙一起过桥到河的左岸，甲单独回到河的右岸将灯带回，然后甲、丙在一起过桥到河的左岸，时间为 $2+1+4=7$ 。

01	<code>#include<iostream></code>
0	<code>#include<cstring></code>
2	<code>using namespace std;</code>
0	<code>const int SIZE=100;</code>
3	<code>const int INFINITY = 10000;</code>
0	<code>const bool LEFT=true;</code>
4	<code>const bool RIGHT=false;</code>

0	const bool LEFT_TO_RIGHT=true;
5	const bool RIGHT_TO_LEFT=false;
0	int n, hour[SIZE];
6	bool pos[SIZE];
07	int max(int a,int b){
0	if(a>b) return a;
8	else return b;
0	}
9	int go(bool stage)
10	{
11	int i,j, num, tmp,ans;
12	if(stage==RIGHT_TO_LEFT)
13	{
14	num=0;
15	ans=0;
16	for(i=1;i<=n;i++)
17	if(pos[i]==RIGHT)
18	{
19	num++;
2	if(hour[i]>ans)
0	ans=hour[i];
21	}

2	if(①)
2	return ans;
2	ans=INFINITY;
3	for(i=1;i<=n-1;i++)
2	if(pos[i]==RIGHT)
4	for(j=i+1;j<=n;j++)
2	if(pos[j]==RIGHT)
5	{
2	pos[i]=LEFT;
6	pos[j]=LEFT;
27	tmp=max(hour[i], hour[j])+②;
2	if(tmp<ans)
8	ans=tmp;
2	pos[i]=RIGHT;
9	pos[j]=RIGHT;
3	}
0	return ans;
31	}
3	if(stage==LEFT_TO_RIGHT)
2	{
3	ans=INFINITY;
3	for(i=1;i<=n;i++)

3	if(③)
4	{
3	pos[i]=RIGHT;
5	tmp=④;
3	if(tmp<ans)
6	ans=tmp;
37	⑤;
3	}
8	return ans;
3	}
9	return 0;
4	}
0	int main()
41	{
4	int i;
2	cin>>n;
4	for(i=1;i<=n;i++)
3	{
4	cin>>hour[i];
4	pos[i]=RIGHT;
4	}
5	cout<<go(RIGHT_TO_LEFT)<<endl;

4	return 0
6	}
47	
4	
8	
4	
9	
5	
0	
51	
5	
2	
5	
3	
5	
4	
5	
5	
5	
6	
57	
5	

8

5

9

6

0

61

6

2

6

3

6

4

6

5

6

6

67

6

8

6

9

70

