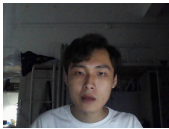


每日一题day17_6月7日测评结果

考生信息



张博翔

考号：1675 | 学校：陕西科技大学 | 邮箱：1761607418@qq.com | 职位：43班 |

参考区域: 陕西省西安市 (111.114.0.2) | 做题用时：01:57:58(2019-06-07 22:25:22 - 2019-06-08 00:23:31)

考生成绩



知识点技能图谱



历史笔试记录

序号	试卷名称	排名	总得分	得分详情	作弊嫌疑	笔试时间
1	每日一题day1_5月20日	4.0%	95.0/100	单选:45.0分 编程:50.0分	否	2019-05-19 16:08:49
2	每日一题day02_5月21日	51.0%	60.0/100	单选:30.0分 编程:30.0分	否	2019-05-20 17:40:56
3	每日一题day03_5月22日	4.0%	95.0/100	单选:45.0分 编程:50.0分	否	2019-05-21 12:59:15
4	每日一题day04_5月23日	8.0%	95.0/100	单选:45.0分 编程:50.0分	否	2019-05-22 10:40:18
5	每日一题day05_5月24日	29.0%	75.0/100	单选:25.0分 编程:50.0分	否	2019-05-22 20:39:16
6	每日一题day06_5月25日	10.0%	95.0/100	单选:45.0分 编程:50.0分	否	2019-05-24 10:41:21
7	每日一题day07_5月27日	6.0%	95.0/100	单选:45.0分 编程:50.0分	否	2019-05-26 15:09:41
8	每日一题day08_5月28日	18.0%	90.0/100	单选:40.0分 编程:50.0分	否	2019-05-27 14:06:31
9	每日一题day09_5月29日	13.0%	90.0/100	单选:40.0分 编程:50.0分	否	2019-05-28 15:11:53

序号	试卷名称	排名	总得分	得分详情	作弊嫌疑	笔试时间
10	每日一题day10_5月30日	6.0%	90.0/100	单选:40.0分 编程:50.0分	否	2019-05-29 11:32:20
11	每日一题day11_5月31日	14.0%	80.0/100	单选:30.0分 编程:50.0分	否	2019-05-30 11:22:09
12	每日一题day12_6月1日	31.0%	75.0/100	单选:25.0分 编程:50.0分	否	2019-05-31 10:33:31
13	每日一题day13_6月3日	9.0%	85.0/100	单选:35.0分 编程:50.0分	否	2019-06-02 15:20:12
14	每日一题day14_6月4日	41.0%	50.0/100	单选:25.0分 编程:25.0分	否	2019-06-03 11:53:13
15	每日一题day15_6月5日	6.0%	85.0/100	单选:35.0分 编程:50.0分	否	2019-06-04 12:02:48
16	每日一题day16_6月6日	6.0%	80.0/100	单选:30.0分 编程:50.0分	否	2019-06-05 16:32:52

编码能力

题号	正确性	提交次数	做题用时	使用语言	运行时间	占用内存	编程思路	代码规范	成绩排名
编程题1	100%	1	00:23:55	C++	5ms	508K	优	优	1%
编程题2	100%	5	01:09:34	C++	6ms	588K	优	优	1%

1 [平均分3.9分 | 87人正确/112人做题 | 用时：2分] 得分：5.0 / 5.0
以下代码

```
char *ptr;
char myString[] = "abcdefg";
ptr = myString;
ptr += 5;
```

代码执行之后ptr指向的内容是？

- A Compiler error
- B f
- C efg
- D defg

他的回答： B (正确)
正确答案： B

2 [平均分4.3分 | 93人正确/109人做题 | 用时：2分] 得分：5.0 / 5.0
已知int a[]={1,2,3,4,5} ; int*p[]={a,a+1,a+2,a+3} ; int **q=p ; 表达式*(p[0]+1)+**(q+2)的值是_____。

- A 5
- B 6
- C 7
- D 8
- E 4
- F 9

他的回答： A (正确)
正确答案： A

3 [平均分4.1分 | 88人正确/108人做题 | 用时：3分] 得分：5.0 / 5.0

以下代码的输出结果是？

```
char *p="abc";
char *q="abc123";
while(*p==*q)
print("%c %c",*p,*q);
```

- A aabbcc
- B aabbcc123
- C abcabc123
- D 代码段错误

他的回答： D (正确)

正确答案： D

4 [平均分2.4分 | 51人正确/108人做题 | 用时：3分 得分：5.0 / 5.0

假设在一个 32 位 little endian 的机器上运行下面的程序，结果是多少？

```
#include <stdio.h>
int main(){
    long long a = 1, b = 2, c = 3;
    printf("%d %d %d\n", a, b, c);
    return 0;
}
```

- A 1,2,3
- B 1,0,2
- C 1,3,2
- D 3,2,1

他的回答： B (正确)

正确答案： B

5 [平均分2.9分 | 59人正确/103人做题 | 用时：2分 得分：5.0 / 5.0

下列给定程序中，函数fun的功能是：求ss所指字符串数组中长度最短的字符串所在的行下标，作为函数值返回，并把其串长放在形参n所指的变量中。ss所指字符串数组中共有M个字符串，且串长小于N。

请在程序的下画线处填入正确的内容并将下画线删除，使程序得出正确的结果。

试题程序。

```
#define M 5
#define N 20
int fun(char(*ss)[N], int *n)
{
    int i, k = 0, len = N;
    for (i = 0; i < _____; i++)
    {
        len = strlen(ss[i]);
        if (i == 0)
            *n = len;
        if (len _____ *n)
        {
            *n = len;
            k = i;
        }
    }
    return ( _____ );
}
main( )
```

```

{
    char ss[M][N] = {"shanghai", "guangzhou", "beijing", "tianjing", "chongqing"};
    int n, k, i;
    printf("\nThe original strings are:\n");
    for (i = 0; i < M; i++)
        puts(ss[i]);
    k = fun(ss, &n);
    printf("\nThe length of shortest string is: %d\n", n);
    printf("\nThe shortest string is: %s\n", ss[k]);
}

```

- A N , < , k
- B N , > , k
- C M , < , k
- D M , > , k

他的回答： C (正确)

正确答案： C

6 [平均分3.6分 | 73人正确/101人做题 | 用时：2分 | 得分：5.0 / 5.0]

调用一成员函数时, 使用动态联编的情况是？

- A 通过对象调用一虚函数
- B 通过指针或引用调用一虚函数
- C 通过对象调用静态函数
- D 通过指针或应用调用一静态函数

他的回答： B (正确)

正确答案： B

7 [平均分1.3分 | 25人正确/98人做题 | 用时：<1分 | 得分：5.0 / 5.0]

如何捕获异常可以使得代码通过编译？

```

class A {
public:
    A(){}
};
void foo(){
    throw new A;
}

```

- A catch (A && x)
- B catch (A * x)
- C catch (A & x)
- D 以上都是

他的回答： B (正确)

正确答案： B

8 [平均分3.7分 | 77人正确/104人做题 | 用时：2分 | 得分：5.0 / 5.0]

下列代码可以通过编译吗？如何修改使其通过编译？

```

template <class T>
struct sum {
    static void foo(T op1 , T op2){
        cout << op1 << op2;
    }
}

```

```
}  
};  
sum::foo(1,3);
```

- A 编译通过
- B 应该去掉static关键字
- C 调用应该如下：sum<int>:: foo(1,3)
- D 调用应该如下：sum:: <int>foo(1,3)

他的回答： C (正确)

正确答案： C

9 [平均分3.3分 | 67人正确/101人做题 | 用时：<1分  得分：0.0 / 5.0

下面这段程序的输出是什么？

```
class A{  
    public:  
        A(){p();}  
        virtual void p(){print("A")}  
        virtual ~A(){p();}  
};  
class B:public A{  
    public:  
        B(){p();}  
        void p(){print("B")}  
        ~B(){p();}  
};  
int main(int, char**){  
    A* a=new B();  
    delete a;  
}
```

- A AABBB
- B BBAA
- C ABAB
- D ABBA

他的回答： B (错误)

正确答案： D

10 [平均分1.2分 | 24人正确/99人做题 | 用时：<1分  得分：0.0 / 5.0

有如下程序段：

```
#include <iostream>  
using namespace std;  
  
class A {  
    public:  
        ~A() {  
            cout << "~A()";  
        }  
};  
class B{  
    public:  
        virtual ~B() {  
            cout << "~B()";  
        }  
};
```

```
}  
};  
class C: public A, public B {  
public:  
    ~C() {  
        cout << "~C()";  
    }  
};  
int main() {  
    C * c = new C;  
    B * b1 = dynamic_cast<B *>(c);  
    A * a2 = dynamic_cast<A *>(b1);  
    delete a2;  
}
```

则程序输出：

- A ~C()~B()~A()
- B ~C()~A()~B()
- C A)B)都有可能
- D 以上都不对

他的回答： B (错误)
正确答案： D

11 [平均分20.7分 | 66人正确/82人做题 | 提交: 1 次 | 得分：25.0 / 25.0
标题：iNOC产品部-杨辉三角的变形 | 时间限制：1秒 | 内存限制：32768K | 语言限制：不限

【iNOC产品部-杨辉三角的变形】

1
1 1 1
1 2 3 2 1
1 3 6 7 6 3 1
1 4 10 16 19 16 10 4 1

以上三角形的数阵，第一行只有一个数1，以下每行的每个数，是恰好是它上面的数，左上角数到右上角的数，3个数之和（如果不存在某个数，认为该数就是0）。求第n行第一个偶数出现的位置。如果没有偶数，则输出-1。例如输入3,则输出2，输入4则输出3。

输入n(n <= 1000000000)
输入描述：

输入一个int整数

输出描述：

输出返回的int值

示例1：
输入

4

输出

3

代码片段

功能实现			代码提交统计			代码执行统计
	TA的	平均		TA的	平均	答案正确：1
总通过率	100%	89%	使用语言	C++		
基本测试用例通过率	6/6 (100%)	90%	做题用时	00:23:55	00:46:07	
边缘测试用例通过率	4/4 (100%)	89%	提交次数	1	4	

代码效率	代码规范及可读性
TA的 参考 运行时间 5ms 1s 占用内存 508K 32768K	代码规范得分 5.0

他的代码：

做题用时: 23 分钟 语言：C++ 运行时间：5ms 占用内存：508K 程序状态：答案正确

```
#include <iostream>
using namespace std;
int main()
{
    int num = 0;
    int arr[] = {2, 3, 2, 4};
    while(cin >> num)
    {
        if(num < 3)
        {
            cout << -1;
            continue;
        }
        cout << arr[(num - 3) % 4] << endl;
    }
}
```

12 [平均分21.1分 | 64人正确/76人做题 | 提交: 5 次 | 得分：25.0 / 25.0]

标题：超长正整数相加 | 时间限制：1秒 | 内存限制：32768K | 语言限制：不限

【超长正整数相加】

请设计一个算法完成两个超长正整数的加法。

接口说明

```
/*
请设计一个算法完成两个超长正整数的加法。
输入参数：
String addend：加数
String augend：被加数
返回值：加法结果
*/
public String AddLongInteger(String addend, String augend)
{
    /*在这里实现功能*/

    return null;
}
```

输入描述：

输入两个字符串数字

输出描述：

输出相加后的结果，string型

示例1：

输入

1

1000

功能实现			代码提交统计			代码执行统计	
总通过率	TA的 100%	平均 84%	使用语言	TA的 C++	平均	格式错误	: 1
基本测试用例通过率	6/6 (100%)	84%	做题用时	01:09:34	00:37:10	答案错误	: 3
边缘测试用例通过率	4/4 (100%)	84%	提交次数	5	4	答案正确	: 1
代码效率					代码规范及可读性		
运行时间	TA的 6ms	参考 1s	代码规范得分 5.0				
占用内存	588K	32768K					

做题用时: 69 分钟 语言: C++ 运行时间: 6ms 占用内存: 588K 程序状态: 答案正确

```
#include <iostream>
#include <string>
#include <stack>
using namespace std;
int main()
{
    string num1;
    string num2;
    while(cin >> num1 >> num2)
    {
        stack<char> res;
        int len = num1.size() < num2.size() ? num1.size() : num2.size();
        int len2 = num1.size() > num2.size() ? num1.size() : num2.size();
        string& longNum = num1.size() > num2.size() ? num1 : num2;
        string& shortNum = num1.size() <= num2.size() ? num1 : num2;
        bool carryFlag = false;
        //cout << len << " " << len2 << endl;
        for(int i = len2 - 1, j = len - 1; j >= 0; i--, j--)
        {
            int curRes;
            if(carryFlag == true)
            {
                curRes = (longNum[i] - '0') + (shortNum[j] - '0') + 1;
                carryFlag = false;
            }
            else
            {
                curRes = (longNum[i] - '0') + (shortNum[j] - '0');
            }
            if(curRes >= 10)
            {
                carryFlag = true;
                curRes %= 10;
            }
        }
    }
}
```



```
        res.push('0' + curRes);
    }
    for(int i = len2 - len - 1; i >= 0; i--)
    {
        int curRes;
        if(carryFlag == true)
        {
            curRes = (longNum[i] - '0') + 1;
            carryFlag = false;
        }
        else
        {
            curRes = (longNum[i] - '0');
        }
        if(curRes >= 10)
        {
            carryFlag = true;
            curRes %= 10;
        }
        res.push('0' + curRes);
    }
    if(carryFlag == true)
    {
        res.push('1');
    }
    while(!res.empty())
    {
        cout << res.top();
        res.pop();
    }
    cout << endl;
}
}
```