

考生手册

考生应当遵守相关法律法规，严守考场纪律。若有违反，将被取消考试资格。

考试方式为本地编程和调试，然后在考试系统中提交代码，最后在考试结束后统一评测。考试共有 5 题，每题 100 分，根据通过的测试数据比例给分。

一、登录系统

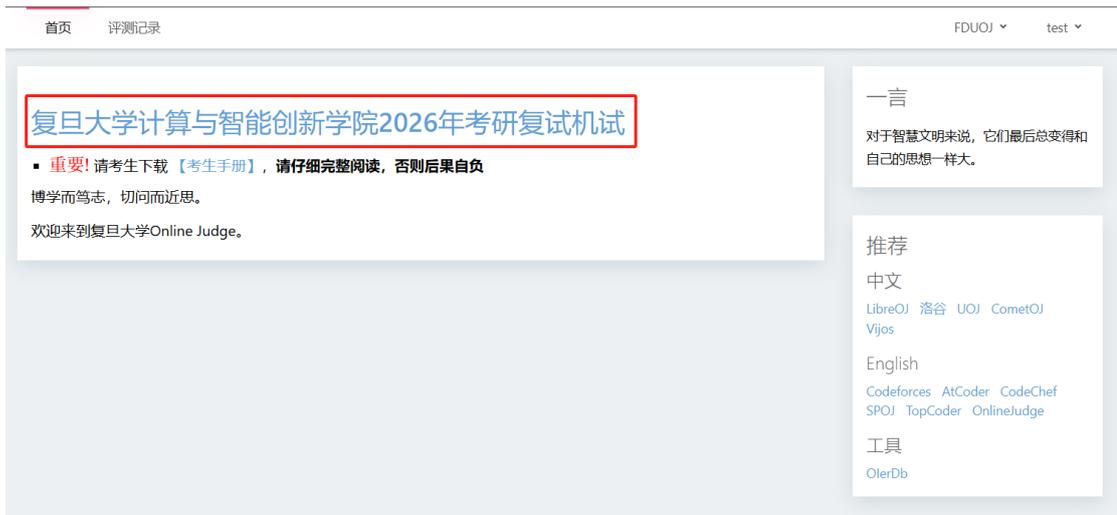
1. 浏览器：请使用 Chrome 或 Edge 浏览器，以正确地访问系统，不要使用 IE 浏览器。如果出现网站连接安全性相关的提示，可以手动点击继续访问来访问。
2. 系统账号与密码：指登录需要的账号和密码，每人具有唯一的一个。
3. 系统访问地址：将在考试前下发。

二、测试流程

1. 访问 OJ 后，点击右上角的登录，输入账号和密码登录账号。

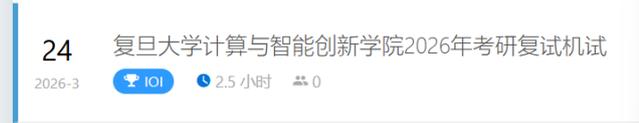


2. 登录后，根据首页提示进入考试域，并时刻保证当前域为该域。如果回到 FDUOJ 域，请重复此操作进入“复旦大学计算与智能创新学院 2026 年考研复试机试”域。

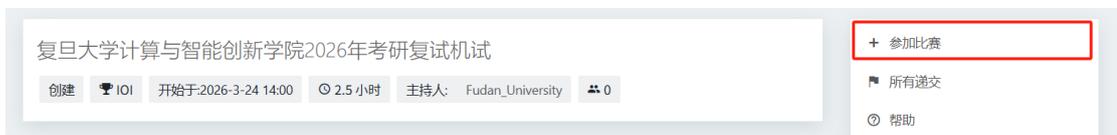


进入考试域后的视图如下。请考生进入考试域页面后下载含有题目 PDF 文件和样例数据的下发文件。下发文件的解压密码将会在考试开始后公布在考试首页。样例中的.in 和.out 文件可以使用 vscode 打开。



3. 点击“比赛”中的  进入比赛。

4. 在比赛详情页点击右上角“**参加比赛**”。



5. 参加比赛后, 请等待考试开始。考试开始后, 点击上方选项卡中的“**题目列表**”查看题目列表, 点击题目名称进入对应的题目描述页面。



题目		
状态	最后提交于	题目
没有提交	-	A A+B Problem

6. 考生需要提交代码时，点击题目描述页面右侧的“提交”按钮。



7. 在提交代码界面选择好代码语言，在下方“代码”输入框里粘贴自己的代码，然后点击“提交”。随后在“题目列表”界面显示样例的评测结果。若考生对同一道题有多次提交，以最后一次提交的代码为准。请保证最后一次提交记录返回的结果不为“编译失败”。

提交以评测

NOTE:
此页面仅用于粘贴代码。
为了获得更好的编辑体验，包括代码高亮和测试运行功能，请返回题目详情页面并点击“进入在线编程模式”按钮。
[忽略 / 不再显示](#)

代码语言

C++

代码语言

C++17(O2)

代码

```
#include <iostream>
using namespace std;
int main(){
    int a, b;
    cin >> a >> b;
    cout << a + b << endl;
    return 0;
}
```

Or upload a file: 未选择任何文件

提交

✓ 100 Pretests Passed

#	状态	分数	耗时	内存占用
#1	✓ Pretests Passed	100	1ms	404 KiB

代码

```
1 #include <iostream>
2 using namespace std;
3 int main(){
4     int a, b;
5     cin >> a >> b;
6     cout << a + b << endl;
7     return 0;
8 }
```

- 考试时间截止后，系统不再接受所有新的提交，请考生注意时间。
- 考生代码下载。对于已提交到考试系统中的代码，如有需要可以在考试系统中查看和下载。具体操作方式为：在比赛页面题目列表的下方的“递交”区域查看。找到对应的提交记录进行查看，并根据需要复制到本地。通过提交到考试系统的方式，可以避免因为机器故障导致考生代码丢失。**因此建议考生合理安排时间提交代码到考试系统，避免因为机器故障带来大的影响。**

状态	题目
✓ 100 Pretests Passed	PO A+B Problem
✓ 100 Pretests Passed	PO A+B Problem

✓ 100 Pretests Passed

#	状态	分数	耗时	内存占用
#1	✓ Pretests Passed	100	1ms	404 KiB

代码 下载

```

1 #include<bits/stdc++.h>
2
3 using namespace std;
4 int a,b;
5 int main(){
6     scanf("%d%d",&a,&b);
7     printf("%d\n",a+b);
8     return 0;
9 }

```

Copy

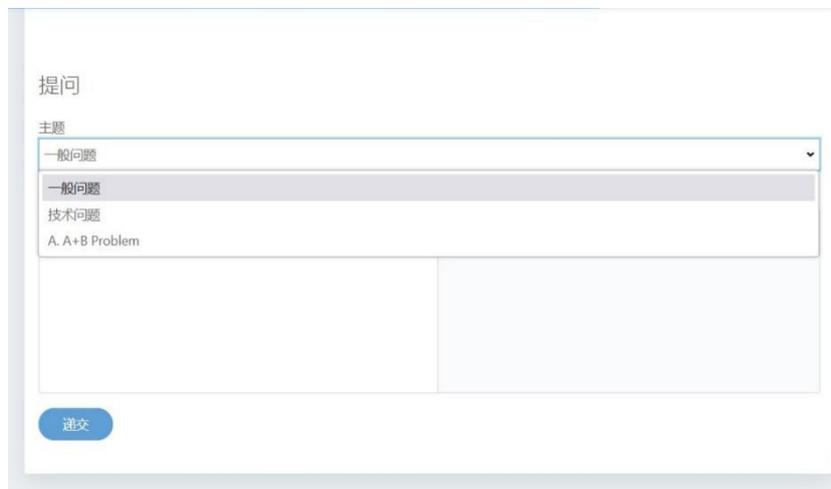
- 每一道题目 C++代码提交的时间限制和空间限制会显示在题目标题下方。由于 java 相对 C++性能较差，在 OJ 中 java 代码的提交时间限制和内存限制均为 C++的两倍。

#A. A+B Problem A

传统题 🕒 1000ms 📄 256MiB

三、 注意事项

1. 考试时每题提供若干组数据用于帮助考生测试，通过这些数据不代表考生程序完全正确。考试系统中返回的结果仅用于帮助考生评判是否有编译错误，和实际得分无任何关联，赛后的最终评测数据中不包括上述数据。
2. 考试的编程语言为 C, C++, Java。
3. 所有程序**必须使用标准输入输出 (stdin, stdout) 进行数据读写，不得使用文件输入输出方法**。请严格按照输出格式进行输出以避免出现错误，输出时每行最后是否有空格以及最后是否有一行空行不影响评测结果。**测试样例通过情况不代表算法一定正确，OJ 显示 Accepted 不代表真正通过题目。**
4. 考试中的公告将在主页更新，且监考员将提醒有新的重要信息，请考生及时刷新页面查看。
5. 和考试机器有关的提问，请直接向监考老师提出。
6. **和考题有关的提问，请使用 OJ 中的“提问”功能提出**，勿就此向监考老师询问；**最后三十分钟将不再接收提问**。提问功能在“题目列表”页面的下端，如下图所示。主题分别对应：不明确问题分类；考试系统使用相关的问题；考题相关的问题。



7. 编译参数:

C: `gcc -Wall --std=c99 -o foo foo.c -lm -DONLINE_JUDGE`

C++: `g++ -Wall -std=c++xx -o foo foo.cc -lm -O2 -I/include -DONLINE_JUDGE`

对于 C++98、C++11、C++14、C++17, 其中的 `-std` 参数分别替换为 `c++98`、`c++11`、`c++14`、`c++17`。

Java: `javac -encoding utf8 ./Main.java`

8. 其他注意事项:

- 1) 保留字: 请不要使用保留字作为标识符, 也请注意不要和库文件中的标识符重名 (比如在使用 C++ 的 `math` 库时, 使用 `y0`, `y1` 作为变量名)。
- 2) C++ `Int64` 如何使用: 请使用 `long long` 进行声明, 并且通过 `cin/cout` 或 `%lld` 读写, 请不要使用诸如 `__int64` 等非标准语法, 否则可能导致运行时错误或编译错误。
- 3) 非标准头文件或者库: 请不要使用诸如 `windows.h` 等部分平台独有的非标准头文件或者库, 否则可能导致编译错误。
- 4) 大规模输入输出: 在大规模输入输出的情况下, 应当选择效率更好的输入输出方法。对于 Java, 请使用下面列出的快速读入方式; 对于 C++, 请使用 `scanf/printf` 或示例代码中快速读入方式。
- 5) 网络拥塞: 当短时间内有大量用户提交时, 可能会排队编译而导致等待时间较长, 特别是在考试即将结束的时候。请考生合理安排时间提交。**因为网络拥塞未能在考试结束前提交成功的, 责任由考生自负。**
- 6) 提交次数限制: 考生请勿在短时间内多次提交代码, 系统将限制**每位考生每分钟仅能提交 2 次代码**, 请考生注意合理安排时间提交。
- 7) 不保证 OJ 的在线编程功能可以正常使用, 请使用本地 IDE 编程后按照考生手册说明提交代码。

9. 示例代码

1) C 代码示例

```
#include <stdio.h>
int main(void) {
    int a, b;
    scanf("%d %d", &a, &b);
    printf("%d", a + b);
    return 0;
}
```

2) C++代码示例

```
#include <iostream>
using namespace std;
int main(){
    int a, b;
    cin >> a >> b;
    cout << a + b << endl;
    return 0;
}
```

3) C++快速读入示例

```

template<class T>
inline bool scan_d(T &ret){
    char c;
    int sgn;
    if (c = getchar(), c == EOF)
        return 0; //EOF
    while (c != '-' && (c < '0' || c > '9')) c = getchar();
    sgn = (c == '-') ? -1 : 1;
    ret = (c == '-') ? 0 : (c - '0');
    while (c = getchar(), c >= '0' && c <= '9')
        ret = ret * 10 + (c - '0');
    ret *= sgn;
    return 1;
}

```

- 4) Java 代码示例，且 Java 主类的名字必须为 Main

```

import java.util.Scanner;

class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int a = in.nextInt();
        int b = in.nextInt();
        System.out.println(a + b);
    }
}

```

- 5) 请注意对于输入输出规模较大的题目(比如数十万的输入输出规模), Java 代码应当考虑使用效率较高的读写方法。比如 StreamTokenizer, BufferedReader, IOException, InputStreamReader, PrintWriter, OutputStreamWriter 等基于 Buffer 的快速输入输出。本段代码跟上面的 Java 代码都可以解决 A+B Problem。

```
import java.io.StreamTokenizer;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.io.OutputStreamWriter;

class Main {
    public static void main(String[] args) throws IOException
    {
        StreamTokenizer in = new StreamTokenizer(new
BufferedReader(new InputStreamReader(System.in)));
        PrintWriter out = new PrintWriter(new
OutputStreamWriter(System.out));
        in.nextToken();
        int n = (int) in.nval;
        in.nextToken();
        int m = (int) in.nval;
        out.printf("%d", n + m);
        out.flush();
        out.close();
    }
}
```

- 6) 这段代码的作用是读入一个数字 n ，以及 n 个数，然后将 n 个数输出出去。

```
import java.io.StreamTokenizer;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.io.OutputStreamWriter;

class Main {
    public static void main(String[] args) throws IOException
    {
        StreamTokenizer in = new StreamTokenizer(new
BufferedReader(new InputStreamReader(System.in)));
        PrintWriter out = new PrintWriter(new
OutputStreamWriter(System.out));
        in.nextToken();
        int n = (int) in.nval;
        int[] result = new int[n];
        for (int i = 0; i < n; ++i) {
            in.nextToken();
            result[i] = (int) in.nval;
        }
        for (int i = 0; i < n; ++i) {
            out.printf("%d ", result[i]);
        }
        out.flush();
        out.close();
    }
}
```