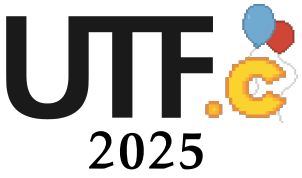
Folha de Informações



13 de dezembro de 2025

https://utfc2025.maratona.c3sl.ufpr.br

Regras gerais

- É permitido consultar, e até mesmo copiar, qualquer volume de material impresso ou manuscrito trazido pelos competidores, como livros, cadernos, e apostilas.
- Não é permitido usar nenhum equipamento eletrônico, nem mesmo celulares ou relógios, exceto a máquina disponibilizada para o time no laboratório.
- Cada time terá uma única máquina para utilizar, independentemente de o time consistir em 1, 2, ou 3 integrantes. Integrantes de um mesmo time poderão, claro, revezar o uso dessa máquina conforme a estratégia que melhor considerarem.
- Não é permitido que competidores de times distintos se comuniquem.
- Se uma pessoa competidora precisar sair do laboratório durante a prova (por exemplo, para ir ao banheiro, ou para comer ou beber alguma coisa), deverá levantar a mão e aguardar o apoio da equipe. Durante esses períodos de curta ausência, essa pessoa não poderá se comunicar com quaisquer pessoas que encontre, a não ser que não pretenda voltar mais ao laboratório.
- Não é permitido consumir alimentos ou bebidas nos laboratórios.
- Toda submissão deve ser uma tentativa legítima de resolver o problema correspondente.
- As impressões devem ser apenas de códigos-fontes submetidos.
- Os competidores poderão pedir esclarecimentos sobre os enunciados dos problemas durante a prova apenas através do recurso `Clarifications`, no sistema de submissão. Não serão respondidas perguntas sobre as soluções dos problemas, apenas sobre os enunciados. Perguntas sobre a prova não poderão ser feitas à equipe de apoio.

Informações do ambiente de testes

- O código-fonte submetido pode ter no máximo 100 kilobytes.
- Os limites de tempo e de memória estão especificados em cada problema.

C/C++

- Comando para compilar C: `gcc -static -O2 -lm <arquivo>.c -o <programa>`.
- Para compilar C++: `g++ -std=c++20 -static -O2 -lm <arquivo>.cpp -o <programa>`.
- Para executar uma solução C/C++: `./<programa>`.
- Entrada e saída em C: `scanf, getchar, printf, putchar`.
- Entrada e saída em C++: as mesmas de C ou os objetos `cout` e `cin`.
- A versão do gcc é 14.2.0.
- Seu programa deve retornar zero, executando, como último comando, `return 0` ou `exit(0)`.
- É sabido que em alguns casos de problemas com entradas muito grandes, os objetos `cin` podem ser lentos, por conta da sincronização de buffers da biblioteca `stdio`. Caso deseje utilizar `cin` e `cout`, um jeito de se contornar este problema é executando o comando `ios_base::sync_with_stdio(0)`, no início de sua função `main`. Note que, neste caso, o uso de `scanf` e `printf` no mesmo programa é contra-indicado, uma vez que, com buffers separados, comportamentos inadequados podem ocorrer.

Java

- A versão do JDK é 21.0.9 e o comando para compilar: `javac <nomerquivo>.java`.

- Para executar uma solução: `java -Xms1024m -Xmx1024m -Xss20m <nomeArquivo>`.
- Entrada e saída: `Scanner`, `BufferedReader`, `BufferedWriter` e `System.out.println`.
- Não declare "`package`" no seu código-fonte.
- Recomenda-se utilizar a convenção de nomear o arquivo como `A.java` e ter uma classe pública `A` de acordo com a letra do problema (A, B, C, ...).
- Cuidado com saídas muito grandes, os métodos de impressão `System.out.print` e `System.out.println` são extremamente lentos. Recomenda-se criar uma string usando um `StringBuilder` e imprimir tudo de uma vez só.

Python

- O Python está na versão 3.13.5 e em erro de sintaxe, será retornado `Runtime Error`.
- Não é garantido que soluções em Python conseguirão executar dentro do tempo limite alocado, porém todos os problemas foram testados com soluções em Python.
- Comando para executar uma solução: `python3 <nome_do_arquivo>.py`.
- Entrada e saída: `input`, `print`, `sys.stdin.read`, `sys.stdin.readline`, `sys.stdin.readlines`, `sys.stdout.write`.

Instruções do sistema de submissão Juĝisto

Tudo no Juĝisto é feito pela interface web, então antes de fazer qualquer ação certifique-se que o navegador está aberto e na URL que se encontra na primeira página desta folha de informações.

Utilizando o sistema

- Aba `Problems` para visualizar cada um dos problema da prova com seus limites de tempo e de memória. Esta página mostra o resultado da sua última submissão e também a quantidade de vezes que o problema foi resolvido por todos os times.
- Aba `Runs` para enviar uma solução para um problema. Escolha o problema apropriado, a linguagem utilizada e envie o arquivo fonte.
- Aba `Clarifications` para solicitar esclarecimentos sobre o enunciado de um problema. Escolha o problema apropriado, escreva sua dúvida e submeta.
- Aba `Score` para ver o placar.
- Aba `Tasks` para enviar tarefas de impressão.
- Aba `Backup` para acessar um ambiente de compartilhamento de arquivos do seu time. Use este ambiente para não perder seus arquivos ao trocar de máquina durante a competição.

Resultado da submissão

- O código será julgado por um juiz automático que avaliará o comportamento do programa para um conjunto de casos de teste secretos. Apesar de cada problema conter exemplos de casos de teste, com os quais os competidores podem testar seus códigos localmente, frisa-se que o conjunto de casos de teste secretos do juiz é geralmente muito maior.
- Cada submissão pode resultar num dos seguintes vereditos:
  - NO - COMPILATION ERROR: seu código não foi aceito porque o compilador não conseguiu compilá-lo corretamente;
  - NO - TIME LIMIT EXCEEDED: seu código não foi aceito porque o programa demorou muito tempo para rodar para algum caso de teste;
  - NO - RUNTIME ERROR: seu código não foi aceito porque o programa foi abortado pelo sistema operacional por executar alguma operação inválida, como acesso indevido à memória ou exceção de ponto flutuante;
  - NO - MEMORY LIMIT EXCEEDED: seu código não foi aceito porque o programa tentou alocar muita memória, algumas vezes isso pode resultar em RUNTIME ERROR;
  - NO - WRONG ANSWER: seu código não foi aceito porque o programa não imprimiu a saída esperada para algum caso de teste;
  - HIDDEN VERDICT: o veredito está escondido porque se iniciou o período de blind;
  - YES: seu código foi aceito e sua equipe ganhou um balão! Parabéns!

Exemplo de problema

Um aluno A quer saber se passou em uma matéria considerando suas notas em duas provas N e M. A média final F é calculada por $F = \frac{N+M}{2}$ e para passar, é necessário ter nota 7 ou mais.

Entrada

A única linha da entrada contém o nome do aluno A e dois inteiros N e M ($1 \leq N, M \leq 10$), que representam a nota do aluno nas duas provas, respectivamente. O nome do aluno é composto apenas de letras latinas minúsculas e tem no máximo 20 caracteres.

Saída

Imprima "`A passou`" se A passou, ou "`F nao e suficiente`" se A não passou (sem as aspas). O valor real F deve ser impresso com exatamente duas casas decimais.

Solução em C

```c
#include <stdio.h>
int main() {
    char A[21]; int N, M;
    scanf("%s %d %d", A, &N, &M);
    double F = (N+M)/2.0;
    if (F >= 7) { printf("%s passou\n", A); }
    else { printf("%0.2f nao e suficiente\n", F); }
    return 0;
}
```

Solução em C++

```cpp
#include <bits/stdc++.h>
using namespace std;
int main() {
    cin.tie(0)->sync_with_stdio(0);
    string A; int N, M;
    cin >> A >> N >> M;
    double F = (N+M)/2.0;
    if (F >= 7) { cout << A << " passou\n"; }
    else { cout << setprecision(2) << fixed << F << " nao e suficiente\n"; }
    return 0;
}
```

Solução em Java

```java
import java.util.Scanner;
import java.util.Locale;
public class A {
    public static void main() {
        Scanner scanner = new Scanner(System.in);
        String A = scanner.next();
        int N = scanner.nextInt(); int M = scanner.nextInt();
        double F = (N+M)/2.0;
        if (F >= 7) { System.out.println(A + " passou"); }
        else { System.out.printf(Locale.US, "%.2f nao e suficiente\n", F); }
    }
}
```

Solução em Python

```python
A, N, M = input().split()
N = int(N)
M = int(M)
F = (N+M)/2
if F >= 7: print(f"{A} passou")
else: print(f"{F:.2f} nao e suficiente")
```