📋 How to Use Input Files for dekodeX Problems

Welcome to **dekodeX**! Our problems use large input files that would be impractical to type manually. This guide will show you exactly how to work with these input files, both online and locally.

🔥 Quick Start Guide

Step 1: Get Your Input File

1. **Download the input file**: On the problem page, click the "Get Input" button
2. **Save the file**: Press `Ctrl + S` on the text file page to download it, or simply copy the content
3. **Name it appropriately**: Save as `input.txt` or `testcase.txt`

Step 2: Choose Your Approach

- 🌐 **Online Compilers** (Recommended for beginners)
- 💻 **Local Development** (Recommended for advanced users)

🌐 Online Compilers (Easy Method)

Recommended Online Platforms

- **OneCompiler**: https://onecompiler.com/cpp
- **CodeChef IDE**: https://www.codechef.com/ide

How to Use

1. Go to your preferred online compiler
2. Paste your code in the editor
3. **Copy the entire content** of your input file
4. **Paste it in the "Input" section** of the online compiler
5. Run your code!

> 💡 ***Pro Tip**: Most online compilers have an "Input" tab or section where you can paste large inputs directly.*

💻 Local Development

For C++ Users

```cpp
#include <iostream>
#include <fstream>
#include <vector>
using namespace std;

int main() {
    // Open the input file
    ifstream inputFile("input.txt");

    if (!inputFile.is_open()) {
        cerr << "Error: Could not open input file!" << endl;
        return 1;
    }

    // Your code here - read from inputFile instead of cin
    int n;
    inputFile >> n;
```

```
    vector<int> numbers(n);
    for (int i = 0; i < n; i++) {
        inputFile >> numbers[i];
    }

    // Close the file
    inputFile.close();

    // Process your data and output the result
    // Your solution logic here...

    return 0;
}
```

For Python Users

```
# Method 1: Using file operations
def solve_from_file():
    with open('input.txt', 'r') as file:
        # Read all lines
        lines = file.readlines()

        # Parse the input
        n = int(lines[0].strip())
        numbers = list(map(int, lines[1].strip().split()))

        # Your solution logic here...

# Method 2: Using sys.stdin redirection
import sys

def solve_with_stdin():
    # Redirect stdin to read from file
    sys.stdin = open('input.txt', 'r')

    # Now use input() normally
    n = int(input())
    numbers = list(map(int, input().split()))

    # Your solution logic here...

# Call your preferred method
solve_from_file()
```

For Java Users

```
import java.io.*;
import java.util.*;

public class Solution {
    public static void main(String[] args) {
        try {
            // Create file reader
```

```
            File file = new File("input.txt");
            Scanner scanner = new Scanner(file);

            // Read input
            int n = scanner.nextInt();
            int[] numbers = new int[n];

            for (int i = 0; i < n; i++) {
                numbers[i] = scanner.nextInt();
            }

            // Close scanner
            scanner.close();

            // Your solution logic here...

        } catch (FileNotFoundException e) {
            System.out.println("Error: input.txt file not found!");
            e.printStackTrace();
        }
    }
}
```

⚠️ Common Issues & Solutions

❌ "File not found" error

- **Solution**: Make sure `input.txt` is in the same directory as your code file
- **Check**: Use `ls` (Linux/Mac) or `dir` (Windows) to list files in the directory

❌ Wrong output format

- **Solution**: Check if you're printing extra spaces or newlines
- **Tip**: Most problems expect just the answer, nothing else

❌ Runtime errors with large inputs

- **Solution**: Use appropriate data types (e.g., `long long` in C++ for large numbers)
- **Tip**: Check the constraints in the problem statement

❌ Online compiler times out

- **Solution**: Your algorithm might be too slow; optimize your approach
- **Alternative**: Try a different online compiler

🆘 Still Need Help?

If you're still having trouble:

Contact us with the option provided on problem page

Happy Coding! 🚀