

# A system to teach Introductory Programming



# People...

- ◆ Rajdeep Das (IITK)
- ◆ Umair Z Ahmed (IITK)
- ◆ Naman Bansal (IITK)
- ◆ Ivan Radiček (VUT)
- ◆ Several Interns
- ◆ Sumit Gulwani (MSR, IITK)
- ◆ Amey Karkare (IITK)
- ◆ More students are added ...

# Challenges

- ◆ Teaching 1<sup>st</sup> level programming course is difficult
- ◆ Specially in India
  - ◆ Lack of proper infrastructure
  - ◆ Many Languages (not the programming languages)
- ◆ Wide gap in the level of exposure to computers
  - ◆ Competitive-programmers to  
Not-even-seen-a-computer-yet!

# Major Challenges for us

- ◆ How to keep good programmers engaged while teaching simple stuff to beginners?
- ◆ How to provide early feedback?
- ◆ Without too many extra resources
  - Limited TA support
  - Varying expertise of TAs
  - Nobody likes to work extra hours ☹
- ◆ Use of existing computers
  - Various flavors of OSes

# Solution?

# Motivation!

## ◆ Code Hunt ?

- Talk with Judith in **Sept 2013** to use it at IIT Kanpur
- Did not work out (for several different reasons)

## ◆ IDEone ?

- Browser based IDE
- Nothing else!

# Setup

- ◆ ESC101 @ IIT Kanpur
- ◆ Approx 400 students every semester
- ◆ Weekly Load
  - for student: 3 Lectures, 1 Tutorial, 1 Lab
  - for instructor: 3 Lectures, 1 Tutorial (summarize the week, do-s and don't-s, quizzes, ...), 4 Labs (?)
- ◆ Other courses have labs too!
  - Space and time table issues => class divided into 4 groups with different lab days
  - ESC101 lab on M, Tu, W, Th

# The System : Finally

◆ Its demo time



Search



# Intelligent Tutoring System

ESC101

karkare@gmail.com

.....

Login



The background is a light blue grid. There are several blue lines: a vertical line on the left, a horizontal line near the top, and another horizontal line near the bottom. There are also blue corner ornaments: a small circle at the top-left intersection of the first vertical and horizontal lines, and a larger circle at the bottom-right intersection of the second horizontal line and the left vertical line.

# Student Interface

# Dashboard

Intelligent Tutoring System

ome

Search

☆

📁

↓

🏠

📶

💬

☰

ITS-Ext | Sigpact ITS

Intelligent Tutoring System ESC101

CodeBook Practice Scratchpad Amey Karkare

🚩 Ongoing Event

TEST-1

Ends on Sat Feb 21 2015 at 12:00:00

Q1	20 Points	not-submitted	<a href="#">🔧 Start Coding</a>
Q2	20 Points	not-submitted	<a href="#">🔧 Start Coding</a>
Q3	20 Points	not-submitted	<a href="#">🔧 Start Coding</a>

📋 GRADE CARD

📄

📊 Course Statistics

Course Problems

Submitted	0
Not Submitted	43

Course Events

Labs	5
Exams	2
Quizzes	0

# Coding Arena

Intelligent Tutoring System

ITS ESC101 File Run Debug Help

CodeBook Practice Scratchpad Amey Karkare

## TEST-1 - Q1 (20 Points)

⌚ Ends in 11 days

↑ Upgrade ↓ Downgrade

### Problem Statement

Write a program to check whether a given number is a **Palindrome** or not.  
An Palindrome number is the one in which the reverse of the number is equal to the number itself.

**Input:** An int, i.e. the number  $n$  ( $n > 0$ )

**Output:**  
YES if the given number is an Palindrome.  
NO otherwise.

**EXAMPLES**

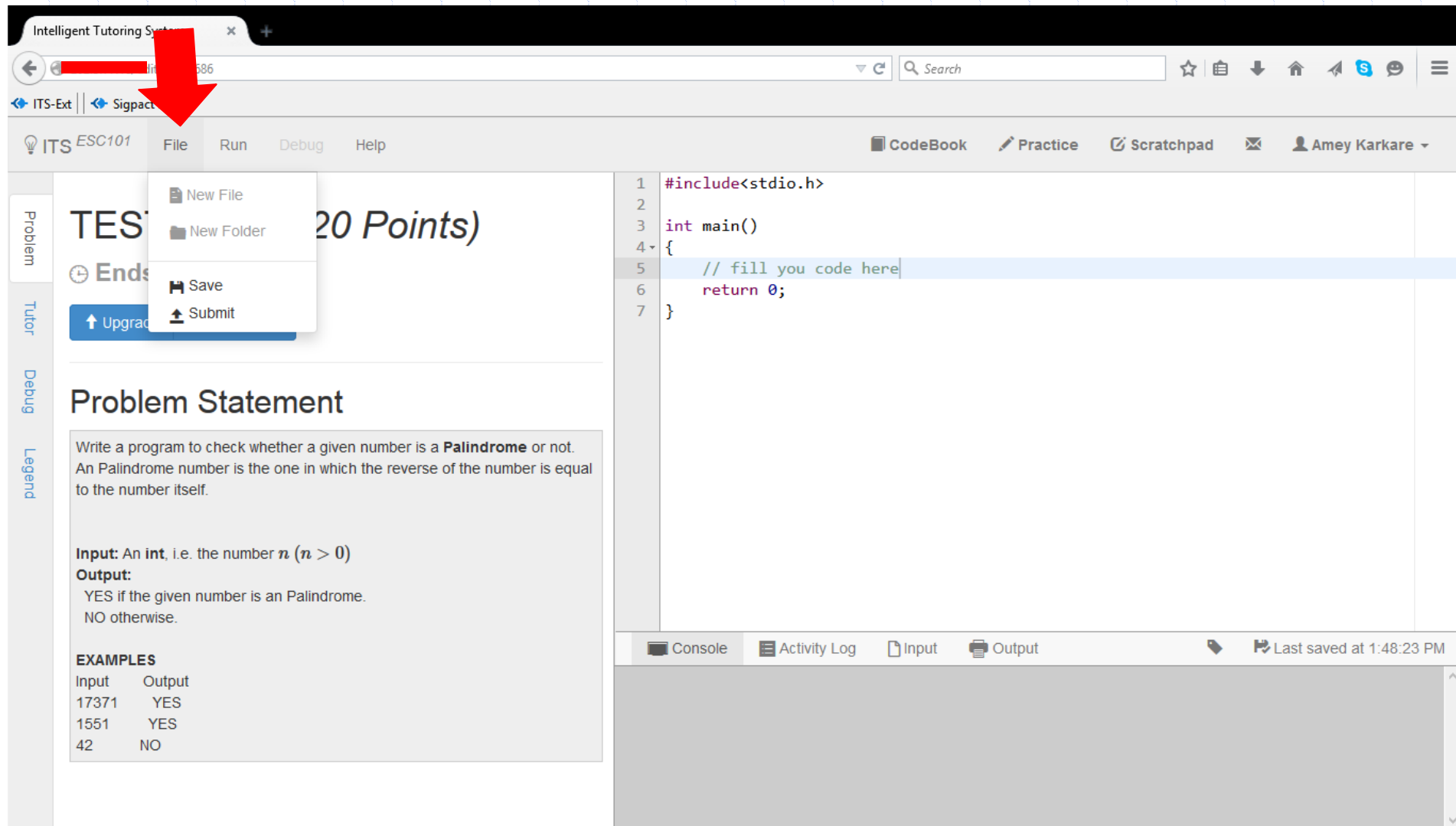
Input	Output
17371	YES
1551	YES
42	NO

```
1 #include<stdio.h>
2
3 int main()
4 {
5     // fill you code here
6     return 0;
7 }
```

Console Activity Log Input Output

Last saved at 1:48:23 PM

# File Menu



The screenshot shows the ITS ESC101 interface. A red arrow points to the 'File' menu in the top navigation bar. The 'File' menu is open, displaying options: 'New File', 'New Folder', 'Save', and 'Submit'. The main content area is divided into three sections: 'Problem Statement', 'Input/Output', and 'EXAMPLES'.

**Problem Statement**

Write a program to check whether a given number is a **Palindrome** or not. An Palindrome number is the one in which the reverse of the number is equal to the number itself.

**Input:** An int, i.e. the number  $n$  ( $n > 0$ )

**Output:**  
YES if the given number is an Palindrome.  
NO otherwise.

**EXAMPLES**

Input	Output
17371	YES
1551	YES
42	NO

The code editor on the right shows a C program template:

```
1 #include<stdio.h>
2
3 int main()
4 {
5     // fill you code here
6     return 0;
7 }
```

The bottom status bar indicates 'Last saved at 1:48:23 PM'.

# Run Menu

The screenshot shows the Intelligent Tutoring System (ITS) interface. A red arrow points to the 'Run' menu in the top navigation bar. The 'Run' menu is open, showing three options: 'Compile', 'Execute', and 'Evaluate'. The 'Evaluate' option is checked. The main content area displays a problem statement for 'TEST-1' (Points) with a timer 'Ends in 1' and buttons for 'Upgrade' and 'Downgrade'. The problem statement asks to write a program to check if a given number is a Palindrome. The input is an integer  $n$  ( $n > 0$ ). The output is YES if the number is a Palindrome, and NO otherwise. Examples are provided: 17371 (YES), 1551 (YES), and 42 (NO). The code editor on the right shows a C program template with a comment to fill in the code. The bottom panel shows the Console, Activity Log, Input, and Output tabs, with a status bar indicating 'Last saved at 1:48:23 PM'.

Intelligent Tutoring System

ITS-Ext | Sigpact ITS

ITS ESC101 File Run Debug Help

CodeBook Practice Scratchpad Amey Karkare

Problem Tutor Debug Legend

TEST-1 (Points)

Ends in 1

Upgrade Downgrade

Problem Statement

Write a program to check whether a given number is a **Palindrome** or not. An Palindrome number is the one in which the reverse of the number is equal to the number itself.

**Input:** An int, i.e. the number  $n$  ( $n > 0$ )

**Output:**  
YES if the given number is an Palindrome.  
NO otherwise.

**EXAMPLES**

Input	Output
17371	YES
1551	YES
42	NO

```
1 #include<stdio.h>
2
3 int main()
4 {
5     // fill you code here
6     return 0;
7 }
```

Console Activity Log Input Output

Last saved at 1:48:23 PM

# Tabs

The screenshot displays the Intelligent Tutoring System (ITS) interface. At the top, a browser tab is labeled "Intelligent Tutoring System". The address bar shows a URL ending in "itor/104686". Below the browser, there's a navigation bar with "ITS-Ext" and "Sigpact ITS". The main interface has a menu bar with "ITS ESC101", "File", "Run", "Debug", and "Help". On the right side of the menu bar, there are links for "CodeBook", "Practice", "Scratchpad", and a user profile "Amey Karkare".

On the left side, there's a vertical sidebar with tabs labeled "Problem", "Tutor", "Debug", and "Legend". The "Legend" tab is currently selected, and a red arrow points to it. The legend lists several error types with corresponding icons:

- ✓ OUTPUT MATCHED
- ✗ OUTPUT DID NOT MATCH
- ⌚ TIME LIMIT EXCEEDED
- ⚠ RUNTIME ERROR
- ⊗ ABNORMAL TERMINATION
- 🔒 RESTRICTED FUNCTION CALL
- 💻 MEMORY LIMIT EXCEEDED

The main area of the interface is a code editor showing a C program:

```
1 #include<stdio.h>
2
3 int main()
4 {
5     // fill you code here
6     return 0;
7 }
```

At the bottom, there's a status bar with tabs for "Console", "Activity Log", "Input", and "Output". The "Output" tab is active, showing a large empty area. The status bar also indicates "Last saved at 1:48:23 PM".

# Editor Window

Intelligent Tutoring System

itor/104686

Search

ITS-Ext | Sigpact ITS

ITS ESC101 | File | Run | Debug | Help

CodeBook | Practice | Scratchpad | Amey Karkare

Problem

Tutor

Debug

Legend

## TEST-1 - Q1 (20 Points)

🕒 Ends in 11 days

⬆ Upgrade ⬇ Downgrade

### Problem Statement


Write a program to check whether a given number is a **Palindrome** or not.  
An Palindrome number is the one in which the reverse of the number is equal to the number itself.

**Input:** An int, i.e. the number  $n$  ( $n > 0$ )  
**Output:**  
YES if the given number is an Palindrome.  
NO otherwise.

**EXAMPLES**

Input	Output
17371	YES
1551	YES
42	NO

```
1 #include<stdio.h>
2
3 int main()
4 {
5     int num;
6     scanf("%d", &num);
7     return 0;
8 }
```



Console | Activity Log | Input | Output

Last saved at 2:09:46 PM

Compilation failed.  
program.c: In function 'main':  
program.c:5:18: error: 'num' undeclared (first use in this function)  
 scanf("%d", &num);  
 ^  
  
program.c:5:18: note: each undeclared identifier is reported only once for each function it appears in

# Run → Compile

Intelligent Tutoring System

editor/104686

ITS-Ext | Sigpact ITS

ITS ESC101 File Run Debug Help

CodeBook Practice Scratchpad

Amey Karkare

Problem

Tutor

Debug

Legend

## TEST-1 - Q1 (20 Points)

🕒 Ends in 11 days

⬆ Upgrade ⬇ Downgrade

### Problem Statement

Write a program to check whether a given number is a **Palindrome** or not.  
An Palindrome number is the one in which the reverse of the number is equal to the number itself.

**Input:** An int, i.e. the number  $n$  ( $n > 0$ )  
**Output:**  
YES if the given number is an Palindrome.  
NO otherwise.

**EXAMPLES**

Input	Output
17371	YES
1551	YES
42	NO

```
1 #include<stdio.h>
2
3 int main()
4 {
5     scanf("%d", &num);
6     return 0;
7 }
```

Console Activity Log Input Output

Last saved at 1:57:15 PM

Compilation failed.  
program.c: In function 'main':  
program.c:5:18: error: 'num' undeclared (first use in this function)  
 scanf("%d", &num);  
 ^  
  
program.c:5:18: note: each undeclared identifier is reported only once for each function it appears in

🛡️

Your program did NOT compile successfully! Please check the annotations on the editor and/or the compiler messages in the console.



# Run → Compile

The screenshot shows the ITS ESC101 interface. On the left, the 'Problem' tab is active, displaying 'TEST-1 - Q1 (20 Points)' which ends in 11 days. Below this is the 'Problem Statement' section, which asks the user to write a program to check if a number is a palindrome. It includes input/output specifications and examples. On the right, the 'CodeBook' tab is active, showing a C program. A red arrow points to the code editor, which contains the following code:

```
1 #include<stdio.h>
2
3 int main()
4 {
5     scanf("%d", &num);
```

An error message box is displayed over the code, stating: 'You are using a variable over here that is undeclared. 'num' has not been declared before. Please declare this variable as something like 'int num;' or 'int num = 0;'. ' Below the code editor, the 'Console' tab shows the compilation error: 'Compilation failed. program.c: In function 'main': program.c:5:18: error: 'num' undeclared (first use in this function) scanf("%d", &num);'. A note at the bottom states: 'program.c:5:18: note: each undeclared identifier is reported only once for each function it appears in'.

**Problem Statement**

Write a program to check whether a given number is a **Palindrome** or not. An Palindrome number is the one in which the reverse of the number is equal to the number itself.

**Input:** An int, i.e. the number  $n$  ( $n > 0$ )

**Output:**

YES if the given number is an Palindrome.  
NO otherwise.

**EXAMPLES**

Input	Output
17371	YES
1551	YES
42	NO

```
1 #include<stdio.h>
2
3 int main()
4 {
5     scanf("%d", &num);
```

You are using a variable over here that is undeclared. 'num' has not been declared before. Please declare this variable as something like 'int num;' or 'int num = 0;'. ' .

Console Activity Log Input Output

Compilation failed.  
program.c: In function 'main':  
program.c:5:18: error: 'num' undeclared (first use in this function)  
scanf("%d", &num);  
^  
program.c:5:18: note: each undeclared identifier is reported only once for each function it appears in

# Run → Compile

Intelligent Tutoring System

itor/104686

ITS-Ext | Sigpact ITS

ITS ESC101 | File | Run | Debug | Help

CodeBook | Practice | Scratchpad

Amey Karkare

Your program compiled successfully!  
You may view any compiler messages in the console.

Problem

Tutor

Debug

Legend

## TEST-1 - Q1 (20 Points)

🕒 Ends in 11 days

⬆ Upgrade ⬇ Downgrade

### Problem Statement

Write a program to check whether a given number is a **Palindrome** or not. An Palindrome number is the one in which the reverse of the number is equal to the number itself.

**Input:** An int, i.e. the number  $n$  ( $n > 0$ )

**Output:**  
YES if the given number is an Palindrome.  
NO otherwise.

**EXAMPLES**

Input	Output
17371	YES
1551	YES
42	NO

```
#include<stdio.h>

int main()
{
    int num;
    scanf("%d", |num);
}
```

When you use format specifiers such as '%d', the type of corresponding variable must match. It is also likely that you forgot to use addressof (&) in scanf.

You cannot use a 'int' variable over here as '%d' requires a 'int \*' type. You should always initialize a variable before you use it in your program. Uninitialized variables tend to contain garbage values and your program may not produce the output that you desired. The variable 'num' is uninitialized over here. Set it to something like 'num=0'.

Console | Activity Log | Input | Output

Last saved at 2:09:46 PM

# Run → Compile

Intelligent Tutoring System

itor/104686

ITS-Ext | Sigpact ITS

ITS ESC101 File Run Debug Help

CodeBook Practice Scratchpad Amey Karkare

Problem

Tutor

Debug

Legend

## TEST-1 - Q1 (20 Points)

🕒 Ends in 11 days

⬆ Upgrade ⬇ Downgrade

### Problem Statement

Write a program to check whether a given number is a **Palindrome** or not. An Palindrome number is the one in which the reverse of the number is equal to the number itself.

**Input:** An int, i.e. the number  $n$  ( $n > 0$ )

**Output:**

YES if the given number is an Palindrome.  
NO otherwise.

**EXAMPLES**

Input	Output
17371	YES
1551	YES
42	NO

```
1 #include<stdio.h>
2
3 int main()
4 {
5     float num;
6     scanf("%d", &num);
7
8 }
```

When you use format specifiers such as '%d', the type of corresponding variable must match. It is also likely that you forgot to use addressof (&) in scanf.

You cannot use a 'float \*' variable over here as '%d' requires a 'int \*' type.

Console Activity Log Input Output

Last saved at 2:09:46 PM

# Run → Evaluate; Feedback

The screenshot displays the ITS interface with a C program in the editor, its evaluation results in a table, and feedback messages on the right. Two red arrows point from the feedback messages to the code line containing the undeclared variable 'num'.

**Tutor**

2:14:05 PM

1. You should try to use a loop (for or while) to compute reverse number.

2:14:05 PM

**Evaluation Results**

#	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	
1	17371	YES		✗
2	1551	YES		✗
3	42	NO		✗

Your program passed 0 out of 2 hidden test case(s).  
NOTE: These may not be the only hidden test cases that your program be evaluated upon.

```
1 #include<stdio.h>
2
3 int main()
4 {
5     int num;
6     scanf("%d", &num);
7     return 0;
8 }
```

**Feedback Messages:**

- You have feedback on your program. Look at the tutor tab to view the feedback.
- Your program did not pass all test cases. Please view the tutor panel to get details.

**Console:**

```
Compilation failed.
program.c: In function 'main':
program.c:5:18: error: 'num' undeclared (first use in this function)
    scanf("%d", &num);
                   ^
program.c:5:18: note: each undeclared identifier is reported only once for each function it appears in
```

# Downgrade

Intelligent Tutoring System

itor/104686

ITS-Ext | Sigpact ITS

ITS ESC101 File Run Debug Help

CodeBook Practice Scratchpad Amey Karkare

Problem

Tutor

Debug

Legend

## TEST-1 - Q1 (20 Points)

🕒 Ends in 11 days

↑ Upgrade ↓ Downgrade

### Problem Statement

Write a program to check whether a given number is a **Palindrome** or not.  
An Palindrome number is the one in which the reverse of the number is equal to the number itself.

**Input:** An int, i.e. the number  $n$  ( $n > 0$ )  
**Output:**  
YES if the given number is an Palindrome.  
NO otherwise.

**EXAMPLES**

Input	Output
17371	YES
1551	YES
42	NO

```
1 #include<stdio.h>
2
3 int main()
4 {
5     int num;
6     scanf("%d", &num);
7     return 0;
8 }
```

Console

Activity Log

Input

Output

Compilation failed.

```
program.c: In function 'main':
program.c:5:18: error: 'num' undeclared (first use in this function)
    scanf("%d", &num);
                   ^
program.c:5:18: note: each undeclared identifier is reported only once for each function it appears in
```

Last saved at 2:09:46 PM

# Downgraded Problem

Intelligent Tutoring System

ITS ESC101 File Run Debug Help

CodeBook Practice Scratchpad

Problem Statement

## TEST-1 - Q1 (20 Points)

Ends in 11 days

Upgrade Downgrade

**Problem Statement**

Write a program to print last two digits of a number in reverse. Assume the number to be  $\geq 10$ .

**Input:** An int, i.e. the number  $n(n \geq 10)$ .

**Output:** the reverse of last two digits.

**EXAMPLES**

Input	Output
1234	43
1551	15
42	24
22	22

```
1 #include<stdio.h>
2
3 int main()
4 {
5     int num;
6     scanf("%d", &num);
7     return 0;
8 }
```

Compilation failed.

program.c: In function 'main':

program.c:5:18: error: 'num' undeclared (first use in this function)

```
scanf("%d", &num);
^
```

program.c:5:18: note: each undeclared identifier is reported only once for each function it appears in

Last saved at 2:09:46 PM

# Feedback

Intelligent Tutoring System

itor/104686

ITS-Ext | Sigpact ITS

ITS ESC101 | File | Run | Debug | Help

CodeBook | Practice | Scratchpad | Amey Karkare

Problem


Tutor

Debug

Legend

## Tutor

2:23:43 PM



1. Add assignment to 'd' to BEGINNING of main function.
2. Check loop condition at line 7.
3. Add assignment to 'r' to BEGINNING of loop starting at line 7.
4. Add assignment to 'n' to BEGINNING of loop starting at line 7.
5. Check 'printf'(s) AFTER loop starting at line 7.

2:23:43 PM

### Evaluation Results

#	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	
1	17371	YES		🕒
2	1551	YES		🕒
3	42	NO		🕒

Your program passed 0 out of 2 hidden test

```
1 #include <stdio.h>
2
3 int main() {
4     int n, d, r;
5     r = 0;
6     scanf("%d", &n);
7     while (n >= 0) {
8
9     }
10    if (n == r) printf("YES");
11    else printf("no");
12
13    return 0;
14 }
```

Console | Activity Log | Input | Output

Last saved at 2:24:38 PM

Compilation failed.

program.c: In function 'main':

program.c:5:18: error: 'num' undeclared (first use in this function)

```
scanf("%d", &num);
               ^
```

program.c:5:18: note: each undeclared identifier is reported only once for each function it appears in

# More Feedback

Intelligent Tutoring System

itor/104686

ITS-Ext | Sigpact ITS

ITS ESC101 File Run Debug Help

CodeBook Practice Scratchpad

Amey Karkare

Tutor

2:26:57 PM

1. Use new (int) variable.  
2. Check assignment to 'r' at line 10.  
3. Check 'printf'(s) AFTER loop starting at line 7.

2:26:57 PM

Evaluation Results

#	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	
1	17371	YES	no	✗
2	1551	YES	no	✗
3	42	NO	no	✗

Your program passed 0 out of 2 hidden test

```
1 #include <stdio.h>
2
3 int main() {
4     int n, d, r;
5     r = 0;
6     scanf("%d", &n);
7     while (n > 0) {
8         d = n%10;
9         n = n/10;
10        r = r*10 + d;
11    }
12    if (n == r) printf("YES");
13    else printf("no");
14
15    return 0;
16 }
```

Console Activity Log Input Output

Compilation failed.  
program.c: In function 'main':  
program.c:5:18: error: 'num' undeclared (first use in this function)  
scanf("%d", &num);  
^  
program.c:5:18: note: each undeclared identifier is reported only once for each function it appears in

Last saved at 2:25:56 PM

2:26 PM 2/10/2015

You have feedback on your program. Look at the tutor tab to view the feedback.

Your program did not pass all test cases. Please view the tutor panel to get details.



# Even more Feedback

Intelligent Tutoring System

itor/104686

ITS-Ext | Sigpact ITS

ITS ESC101 File Run Debug Help

CodeBook Practice Scratchpad

Amey Karkare

Problem

Tutor

Debug

Legend

## Tutor

2:28:55 PM

1. Check assignment to 'r' at line 11.

2. Check 'printf(s) AFTER loop starting at line 8.

2:28:55 PM

### Evaluation Results

#	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	
1	17371	YES	YES	✓
2	1551	YES	YES	✓
3	42	NO	no	✗

Your program passed 2 out of 2 hidden test case(s).

NOTE: These may not be the only hidden test cases that your program be evaluated upon.

```
1 #include <stdio.h>
2
3 int main() {
4     int saved, n, d, r;
5     r = 0;
6     scanf("%d", &n);
7     saved = n;
8     while (n > 0) {
9         d = n%10;
10        n = n/10;
11        r = r*10 + d;
12    }
13    if (saved == r) printf("YES");
14    else printf("no");
15
16    return 0;
17 }
```

Console

Activity Log

Input

Output

Compilation failed.  
program.c: In function 'main':  
program.c:5:18: error: 'num' undeclared (first use in this function)  
 scanf("%d", &num);  
 ^  
program.c:5:18: note: each undeclared identifier is reported only once for each function it appears in

Last saved at 2:28:20 PM

You have feedback on your program. Look at the tutor tab to view the feedback.

Your program did not pass all test cases. Please view the tutor panel to get details.

# Success 😊

Intelligent Tutoring System

ITS ESC101 File Run Debug Help

CodeBook Practice Scratchpad

Amey Karkare

2:29:25 PM

## Tutor

### Evaluation Results

#	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	
1	17371	YES	YES	✓
2	1551	YES	YES	✓
3	42	NO	NO	✓

Your program passed 2 out of 2 hidden test case(s).  
NOTE: These may not be the only hidden test cases that your program be evaluated upon.

2:28:55 PM

1. Check assignment to 'r' at line 11.
2. Check 'printf'(s) AFTER loop starting at line 8.

```
1 #include <stdio.h>
2
3 int main() {
4     int saved, n, d, r;
5     r = 0;
6     scanf("%d", &n);
7     saved = n;
8     while (n > 0) {
9         d = n%10;
10        n = n/10;
11        r = r*10 + d;
12    }
13    if (saved == r) printf("YES");
14    else printf("NO");
15
16    return 0;
17 }
```

Console Activity Log Input Output

Compilation failed.  
program.c: In function 'main':  
program.c:5:18: error: 'num' undeclared (first use in this function)  
scanf("%d", &num);  
^  
program.c:5:18: note: each undeclared identifier is reported only once for each function it appears in

Last saved at 2:28:20 PM

The background is a light blue grid. A solid blue horizontal line spans the width of the page. A solid blue vertical line is positioned on the left side. A solid blue horizontal line is positioned below the main text. A solid blue vertical line is positioned on the right side. There are two small blue circles: one on the left vertical line and one on the right vertical line.

# Admin Interface

# Dashboard

Intelligent Tutoring System

←

🔍 Search

☆

📁

↓

🏠

📶

💬

☰

ITS-Ext | Sigpact ITS

Intelligent Tutoring System *ESC101*

☰

✉

Amey Karkare ▾

🏠

📁

📊

🗃

?

Problems

Manage / Upload

👤

Accounts

Manage

📅

Events

Create & Schedule

📊

DataViZ

Visualize

# Problem Management

The screenshot shows a web browser window with the address bar displaying "problems/manage". The page title is "Intelligent Tutoring System ESC101". The user is logged in as "Amey Karkare". The main heading is "Manage Problems", with an "Upload Problems" button. On the left, there is a sidebar with a home icon and a list of items: "Problems", "Accounts", and three icons representing a folder, a bar chart, and a grid. The "Problems" section is active. The main content area has a "Create a Problem" form with fields for "Local (PVD) Identifier" (example: p1v1d1) and "Category" (example: Strings), and a "+ Create" button. Below the form is a list of problem categories: "LAB-0 (INTRO)", "LAB-1 (IO)", and "LAB-2 (CONDITIONALS)". Under "LAB-2 (CONDITIONALS)", there are six sub-items: "P1-V1-D1 (Mon-Vowels)", "P1-V2-D1 (Mon-Interval)", "P2-V1-D1 (Tue-Equality)", "P2-V2-D1 (Tue-2ndMax)", "P3-V1-D1 (Wed-ObtuseTriangle)", and "P3-V2-D1 (Wed-MidChar)". A large text overlay in the center of the page reads "Select A Problem From The Left" with a hand cursor icon pointing to the left sidebar.

ESC101-ITS Admin

problems/manage

Search

ITS-Ext | Sigpact ITS

Intelligent Tutoring System ESC101

Amey Karkare

Manage Problems

Upload Problems

Create a Problem

Local (PVD) Identifier

eg: p1v1d1

Category

eg: Strings

+ Create

LAB-0 (INTRO)

LAB-1 (IO)

LAB-2 (CONDITIONALS)

P1-V1-D1 (Mon-Vowels)

P1-V2-D1 (Mon-Interval)

P2-V1-D1 (Tue-Equality)

P2-V2-D1 (Tue-2ndMax)

P3-V1-D1 (Wed-ObtuseTriangle)

P3-V2-D1 (Wed-MidChar)

Select A Problem From The Left

# Problem Statement

ESC101-ITS Admin

problems/1633

Search

☆ | 📁 | ⬇ | 🏠 | 📶 | 💬 | ☰

ITS-Ext | Sigpact ITS

💡 Intelligent Tutoring System *ESC101*

☰ | ✉ | 👤 Amey Karkare ▾

🏠

Problems

Accounts

📁

📊

🗖

Create a Problem

Local (PVD) Identifier

eg: p1v1d1

Category

eg: Strings

+ Create

▶ 📁 LAB-0 (INTRO)

▶ 📁 LAB-1 (IO)

▶ 📁 LAB-2 (CONDITIONALS)

▾ 📁 LAB-3 (SERIES)

? P1-V1-D1 (Mon-FactorialRange)

? P1-V2-D1 (Mon-VarCipher)

? P2-V1-D1 (Tue-FibRange)

? P2-V2-D1 (Tue-Special)

? P3-V2-D1 (Wed-FacRange)

? P4-V1-D1 (Thu-Palin)

? P4-V1-D2 (Wed-Palin)

? P4-V2-D1 (Thu-FibRange)

▶ 📁 LAB-4 (PATTERN)

Mon-FactorialRange [\[Change Title\]](#)

🗑

P1-V1-D1

( Practice ☐ )

Problem Statement [\[Edit\]](#)

Write a C Program that takes as input a long integer  $k > 0$  and prints the integer  $n > 0$  such that  $n! \leq k < (n+1)!$ . Here  $n!$  means factorial of positive integer  $n$ .  
Note: Use long data type to store the value of  $k$  and  $n!$ .

Examples:

Input:  
1  
Output:  
1

Input:  
5  
Output:  
2

Input:  
1080

# Instructor's Solution

The screenshot displays the ESC101-ITS Admin web application. The browser's address bar shows the URL `problems/1633`. The application header includes the title "Intelligent Tutoring System ESC101" and the user name "Amey Karkare". A left sidebar contains navigation links for "Problems" (selected), "Accounts", and a folder icon. The main content area is titled "Primary Solution Code editor" and contains a C program for calculating factorials. Below the code editor, there are sections for "Solution Specifications" with an "Add Specification" button, and "Initial Template" with an "[Edit]" link.

ESC101-ITS Admin

problems/1633

ITS-Ext | Sigpact ITS

Intelligent Tutoring System ESC101

Amey Karkare

Problems

Accounts

MockLabExam-1

OldLabExam1-Day2

OldTempExam1

Practice-1 (IO)

Practice-2 (CONDITIONALS)

Practice-2.5 (CONDITIONALS LAB)

Practice-3 (SERIES)

Practice-3.5 (SERIES LAB)

Practice-4 (PATTERN)

Practice-5 (ARRAYS)

## Primary Solution Code editor

```
#include<stdio.h>

int main()
{
    int n=1;
    long k, fact = 1;
    scanf("%ld",&k);

    while (k >= fact) {
        n = n+1;
        fact = fact * n;
    }

    printf("%d\n",n-1);
    return 0;
}
```

## Solution Specifications

+ Add Specification

## Initial Template

[Edit]

```
#include <stdio.h>
```

# Specifications and Template

The screenshot displays the ESC101-ITS Admin web application. The browser's address bar shows the URL `problems/1633`. The application's header includes the title "Intelligent Tutoring System ESC101" and a user profile for "Amey Karkare". A left sidebar contains navigation links for "Problems" (active), "Accounts", and other system functions. The main content area is titled "Solution Specifications" and includes a link to "Add Specification". Below this is the "Initial Template" section, which contains a code editor with the following C code:

```
#include <stdio.h>
#include <stdlib.h>

int main() {
    // Fill this area with your code.
    return 0;
}
```

The "Initial Template" section also features an "[Edit]" link. Below the template is the "TEST CASES" section, which is divided into "AUTOMATICALLY GENERATED TEST CASES" and "MANUALLY ADDED TEST CASES". The "AUTOMATICALLY GENERATED TEST CASES" section states: "This problem has no automated test cases. [Add](#) automated/pre-generated test cases to this problem." The "MANUALLY ADDED TEST CASES" section includes a link to "Add a test case manually".



# Testcases: Automated & Manual

ESC101-ITS Admin

problems/1633

Search

☆

📁

↓

🏠

🔍

S

💬

☰

ITS-Ext | Sigpact ITS

Intelligent Tutoring System *ESC101*

☰

✉

Amey Karkare

⬆

🏠

Problems

Accounts

📁

📊

🗃

```
#include <stdio.h>
#include <stdlib.h>

int main() {
    // Fill this area with your code.
    return 0;
}
```

[Edit]

TEST CASES

AUTOMATICALLY GENERATED TEST CASES

This problem has no automated test cases. [Add](#) automated/pre-generated test cases to this problem.



MANUALLY ADDED TEST CASES





+ Add a test case manually

#	INPUT	OUTPUT	VISIBILITY	
1	1	1	visible	⊗
2	5	2	visible	⊗
3	1080	6	visible	⊗
4	523456723	12	visible	⊗
5	6227020800	13	invisible	⊗
6	20922789888000	16	invisible	⊗

# Control Panel and Settings

Intelligent Tutoring System *ESC101*

 Amey Karkare











CONTROL PANEL

### GCC Compiler Options

+ Flag

#### Current Flags


- -static
- -g
- -Wall
- -lm
- -O1
- -funsigned-char
- -Wno-unused-result




### Execution Sandbox Options

#### Quotas

Time

 milli-seconds

Memory

 bytes

Update Quotas

### PLUGINS


#### Strategy Based Feedback

Author: Ivan Radiček


Enabled ☒ Disabled ☐

### Engine Delays


#### Compilation

 milli-seconds

#### Execution

 milli-seconds

#### Evaluation

 milli-seconds






















Update Delays

# Analytics



## DataViZ > Syntactic Analysis

This section shows the syntactic errors which occur in the programs along with their frequencies and other details. The data has been collected from compiler outputs. The right hand side shows the instances of the selected error.

#	Type	Class	Freq	
1		:X1 undeclared (first use in this function)	7498	
2		unused variable :X1 [-Wunused-variable]	7199	
3		:X1 is used uninitialized in this function [-Wuninitialized]	6761	
4		format :X1 expects argument of type :X2, but argument 2 has type :X3 [-Wformat=]	6149	
5		expected :X1 before :X2 token	4473	
6		control reaches end of non-void function [-Wreturn-type]	4217	
7		each undeclared identifier is reported only once for each function it appears in	4160	
8		expected :X1 before :X2	3934	
9		variable :X1 set but not used [-Wunused-but-set-variable]	2604	
10		expected expression before :X1 token	2346	

# Syntactic Feedback

## DataViZ > Syntactic Analysis

This section shows the syntactic errors which occur in the programs along with their frequencies and other details. The data has been collected from compiler outputs. The right hand side shows the instances of the selected error.

#	Type	Class	Freq	
1	!	:X1 undeclared (first use in this function)	7498	✓
2	⚠	unused variable :X1 [-Wunused-variable]	7199	✓
3	⚠	:X1 is used uninitialized in this function [-Wuninitialized]	6761	✓
4	⚠	format :X1 expects argument of type :X2, but argument 2 has type :X3 [-Wformat=]	6149	✓
5	!	expected :X1 before :X2 token	4473	✓
6	⚠	control reaches end of non-void function [-Wreturn-type]	4217	✓
7	!	each undeclared identifier is reported only once for each function it appears in	4160	✓
8	!	expected :X1 before :X2	3934	✓
9	⚠	variable :X1 set but not used [-Wunused-but-set-variable]	2604	✓
10	!	expected expression before :X1 token	2346	✓

### Feedback

You should always initialize a variable before you use it in your program. Uninitialized variables tend to contain garbage values and your program may not produce the output that you desired. The variable ':X1' is uninitialized over here. Set it to something like ':X1=0'.

[Update](#)

Assignment  
**#69337**

[Show / Hide](#)

Assignment

This section shows the syntactic errors which occur in the programs along with their frequencies and other details. The data has been collected from compiler outputs. The right hand side shows the instances of the selected error.

#	Type	Class	Freq	
1	!	:X1 undeclared (first use in this function)	7498	✓
2	⚠	unused variable :X1 [-Wunused-variable]	7199	✓
3	⚠	:X1 is used uninitialized in this function [-Wuninitialized]	6761	✓
4	⚠	format :X1 expects argument of type :X2, but argument 2 has type :X3 [-Wformat=]	6149	✓
5	!	expected :X1 before :X2 token	4473	✓
6	⚠	control reaches end of non-void function [-Wreturn-type]	4217	✓
7	!	each undeclared identifier is reported only once for each function it appears in	4160	✓
8	!	expected :X1 before :X2	3934	✓
9	⚠	variable :X1 set but not used [-Wunused-but-set-variable]	2604	✓
10	!	expected expression before :X1 token	2346	✓
11	⚠	:X1 may be used uninitialized in this function [-Wmaybe-uninitialized]	2302	✗
12	⚠	suggest parentheses around assignment used as truth value [-Wparentheses]	2124	✗

## Feedback

You should always initialize a variable before you use it in your program. Uninitialized variables tend to contain garbage values and your program may not produce the output that you desired. The variable ':X1' is uninitialized over here. Set it to something like ':X1=0'.

Update

### Assignment #69337

Show / Hide

#2475434

'a' is used uninitialized in this function [-Wuninitialized]

### Assignment #84401

Show / Hide

#2481649

'Celcius' is used uninitialized in this function [-Wuninitialized]

# User Submissions



## Submissions

View code  
submissions  
by users

LAB-0

LAB-1

LAB-2

LAB-3

LAB-4

TEST-1

MockLabExam-1

LabExam-1


1



2





3

4

# User Submissions

 Intelligent Tutoring System *ESC101*

 Amey Karkare ▾



Submissions

LAB-0LAB-1LAB-2LAB-3LAB-4TEST-1MockLabExam-1LabExam-1

DAY 1

DAY 2

DAY 3

DAY 4

FILTERS: Graded ☒ Not-Graded ☐ Submitted ☒ Not-Submitted ☒

Q1 #101421	Q2 #101422	Q1 #101423	Q2 #101424	Q1 #101425	Q2 #101426
Q1 #101427	Q2 #101428	Q1 #101429	Q2 #101430	Q1 #101431	Q2 #101432
Q1 #101433	Q2 #101434	Q1 #101435	Q2 #101436	Q1 #101437	Q2 #101438
Q1 #101439	Q2 #101440	Q1 #101501	Q2 #101502	Q1 #101503	Q2 #101504

# Grading Panel



Viewer



## PROBLEM STATEMENT



View Analytics

Assignment ID: 115501

Marks / 30

Comments

Grade

Editor

Submission

Evaluate

Fri Feb 06 2015 16:53:25



Submitted

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main() {
5     long array[1000]; //stores input
6     long value[1000];
7     long count[1000];
8
9     //inputs
10    int n, i;
11    scanf("%d", &n);
12    for(i=0; i<n; i++)
13        scanf("%ld", &array[i]);
14
15    //initial values
16    value[0] = array[0];
17    count[0] = 1;
18    int index = 0; //index for value[] and count[]
19
20    for(i=1; i<n; i++)
21    {
22        if(array[i] == value[index]) //if successive element
23            is same
24            count[index]++;
25        else {
26            index++; //shift
27            value[index] = array[i]; //new value
28            count[index] = 1; //reset count
29        }
30    }
```

on

Fri Feb 06 2015  
16:53:25

#include <stdio.h>  
#include <stdlib.h>  
  
int main() {  
 long array[1000]; //stores input  
 long value[1000];  
 long count[1000];  
}

manually  
saved

on

Fri Feb 06 2015  
16:53:22

#include <stdio.h>  
#include <stdlib.h>  
  
int main() {  
 long array[1000]; //stores input  
 long value[1000];  
 long count[1000];  
}

autosaved

on

Fri Feb 06 2015  
16:52:42

#include <stdio.h>  
#include <stdlib.h>  
  
int main() {  
 long array[1000]; //stores input  
 long value[1000];  
 long count[1000];  
}

autosaved

on

Fri Feb 06 2015  
16:51:40

#include <stdio.h>  
#include <stdlib.h>  
  
int main() {  
 long array[1000]; //stores  
 long value[1000];  
}



# Analytics per Submission



## Assignment #115247

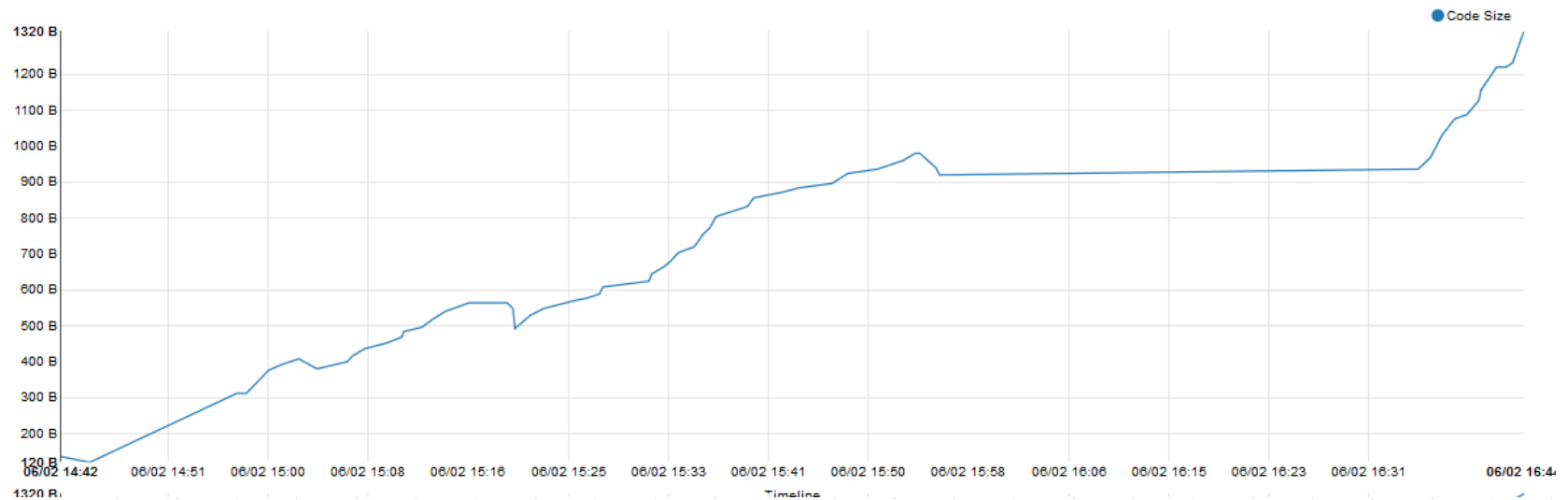
### Analytics

### Summary

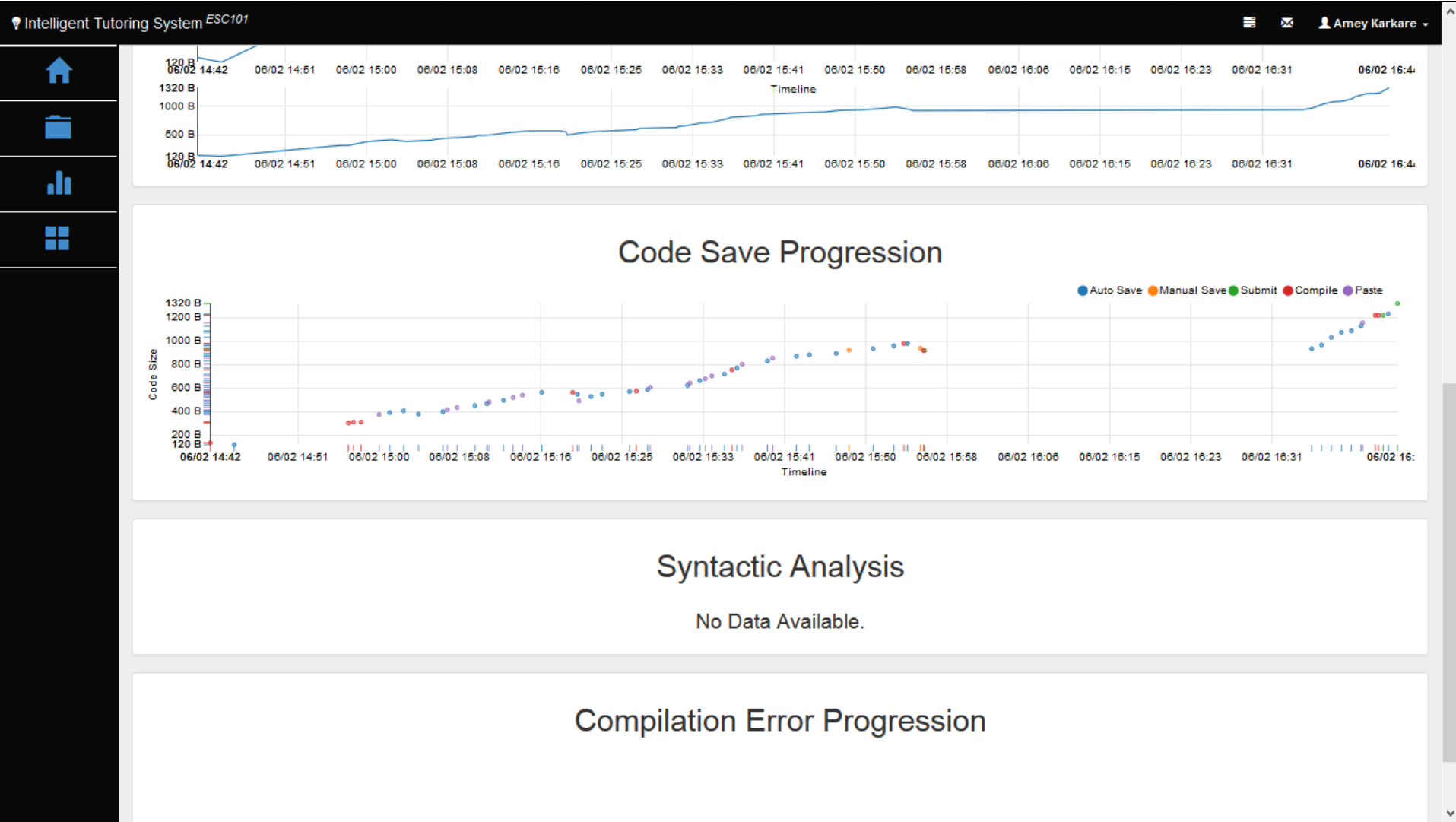
	Executions	Compilations	Submissions
#	1	0	3

Code Size Variation | Code Saves | Syntactic Analysis

### Code Size Variation



# Analytics per Submission



# Concluding Remarks

- ◆ < 9 months old system, Still in experimentation phase
- ◆ Framework allows us to plug-and-play different components
  - Compiler: gcc, python, haskell
  - Feedback: Ivan's strategy based feedback, compiler message rewriting, some ad-hoc scripts
  - Automated Problem Generation: Ad-hoc programs
  - Automated Testcase Generation : KLEE

# Future Work

- ◆ Lots of Data waiting to be processed
- ◆ Many HCI issues to be resolved
- ◆ Feedback tool has known limitations
  - Response time
  - False positives
  - Large # of specifications
- ◆ Crowdsourcing
  - test cases, peer review, specification generation, ...



Thank you.  
Questions?



# Some References

- ◆ Massively Empowered Classroom: Enhancing Technical Education in India (MSR-TR-2013-127)
- ◆ *Feedback Generation for Performance Problems in Introductory Programming Assignments*, FSE 2014, Sumit Gulwani, Ivan Radiček, Florian Zuleger
- ◆ KLEE: Unassisted and Automatic Generation of High-Coverage Tests for Complex Systems Programs, OSDI 2008, Cristian Cadar, Daniel Dunbar, Dawson Engler