



A Web-based automatic C/C++ program evaluator

Project by
Manjela Toppo (CS16MTECH11007)
Rashmi HTI (CS16MTECH11013)
Sherin Thomas (CS16MTECH11016)

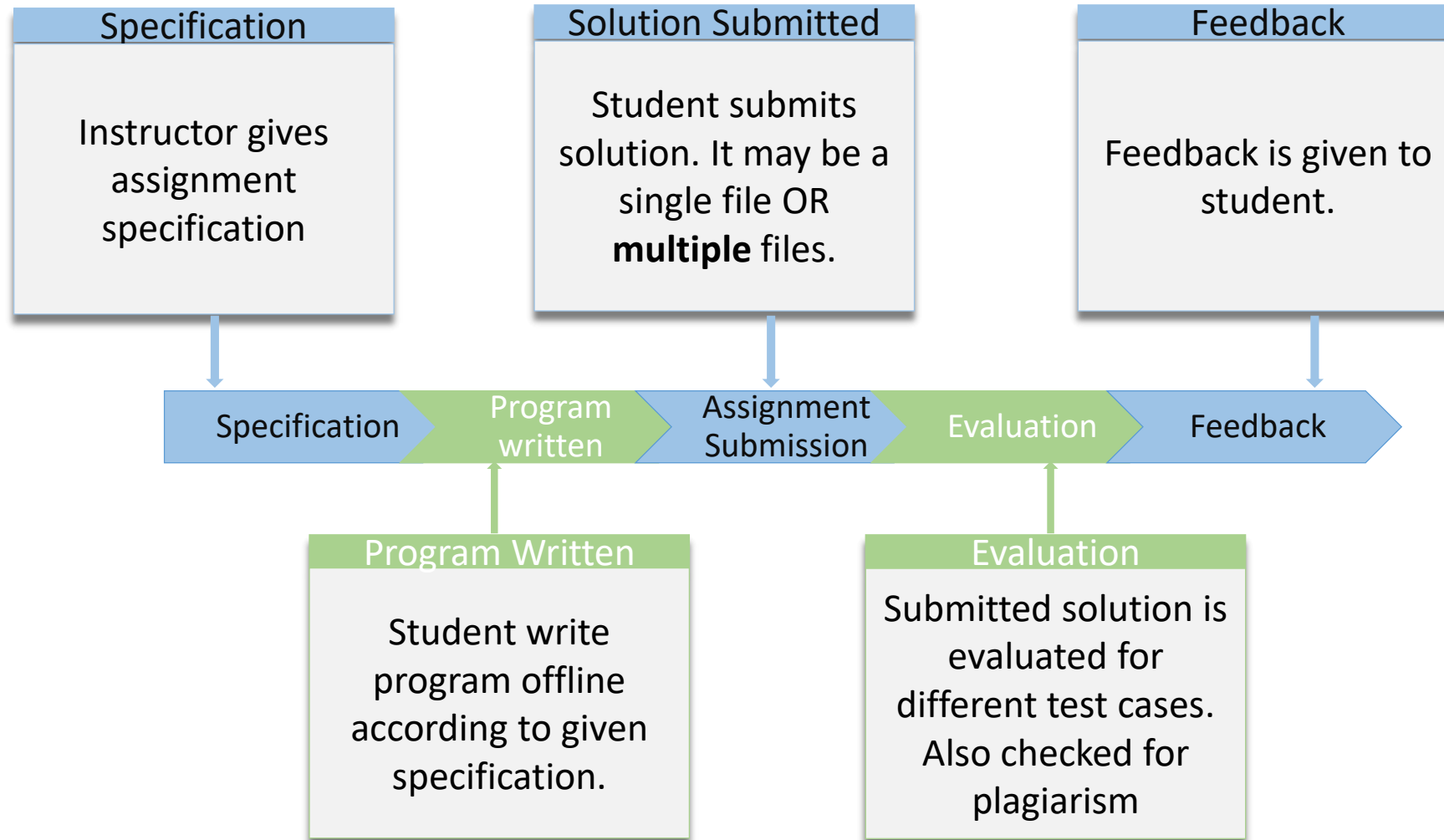
Outline

- Context
- Assignment workflow
- Evaluation process
- Limitations of existing systems
- C-Judge
 - Express Web Framework
 - Architecture
 - Design challenges
 - Future work

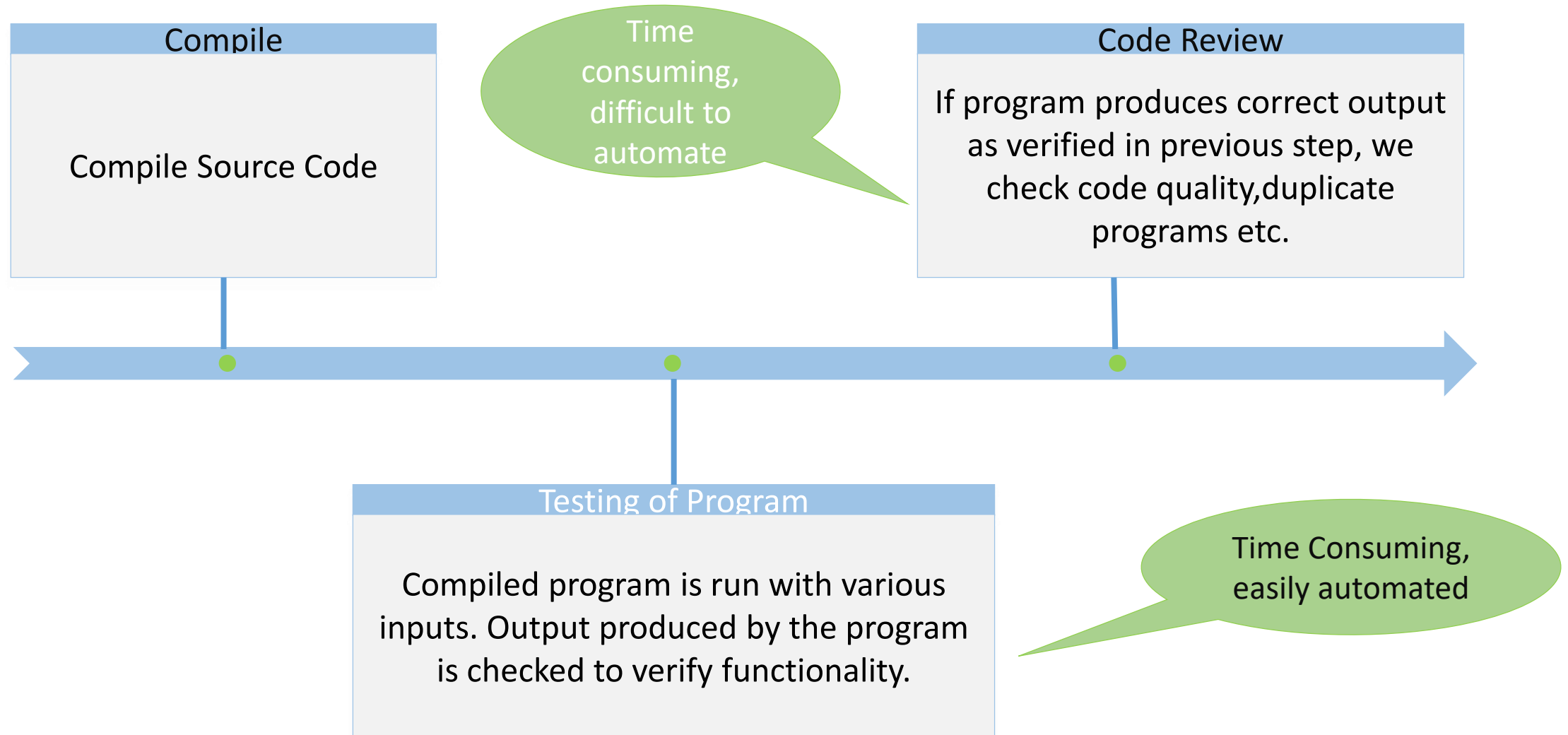
Context

- Course on programming or advanced programming
- Such courses can have large (100+) enrollment
- A course may involve multiple programming assignments
- Each assignment may contain multiple files
- Flexibility in program evaluation and feedback generation is needed

Assignment workflow



Evaluation Process



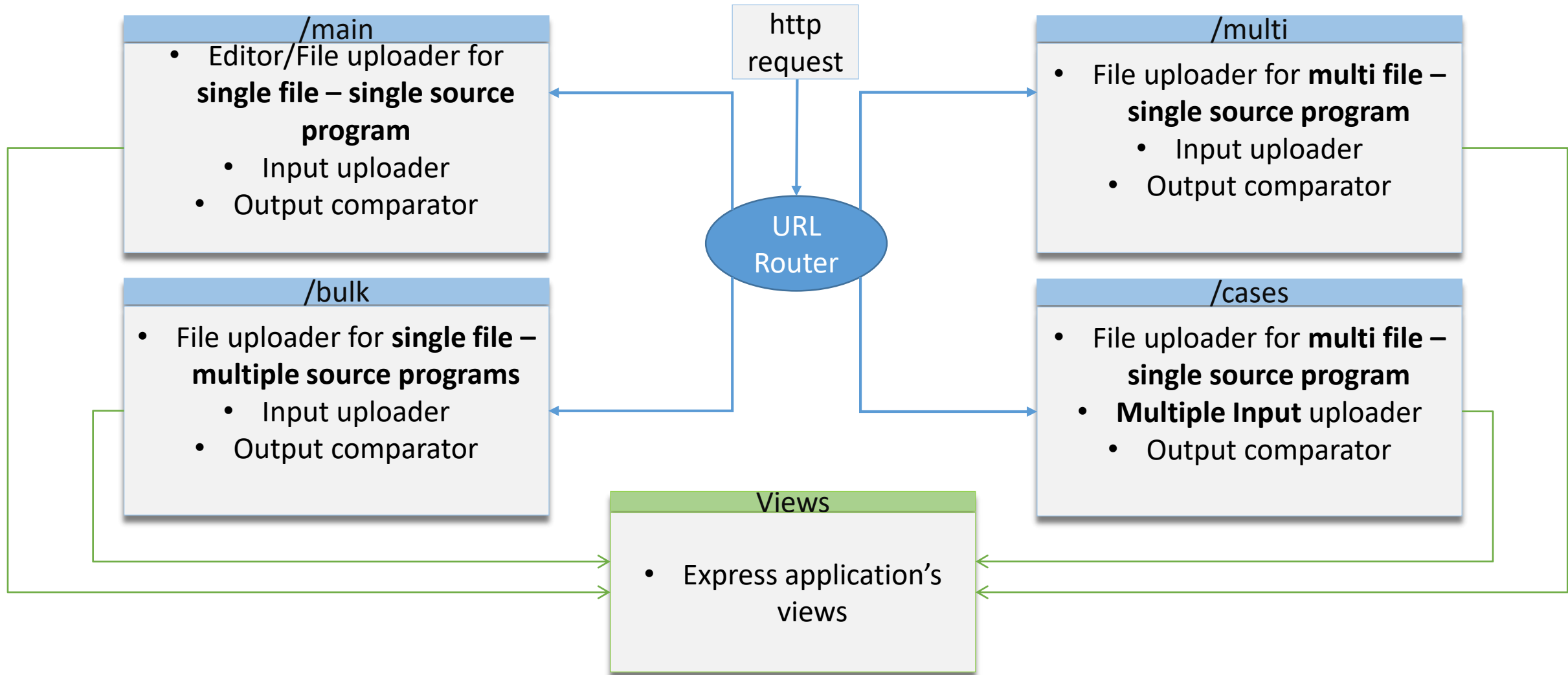
Limitations of Existing Systems

- Existing systems
 - hakerrank.com
 - codechef.com
 - SPOJ (Sphere Online Judge)
 - topcoder.com
- Accepts only single file of source code
- Cannot be used to grade assignments with multiple source files
- Feedback provided by them is very limited

Express Web Framework

- Written in JavaScript
- Server-side web and mobile application framework for Node.js
- Lightweight framework with extra, built-in web application features
- Templating engines
 - Jade and EJS, which facilitate the flow of data into a website structure
- Operating system
 - It's cross-platform, so it's not limited to one OS

C-Judge Architecture



Single File-Single source code Editor/File Upload

C-JUDGE
C/C++ PROGRAM EVALUATOR

HOME

MULTIPLE TEST CASES

MULTIPLE-FILE PROGRAM

MULTIPLE SOURCE PROGRAMS

CODE EDITOR

```
#include<stdio.h>

int main()
{
    int a , b, c ;

    scanf("%d",&a);
    scanf("%d",&b);
    c = a + b ;
    printf("%d",c);
    return 0;
}
```

INPUT

4 5

EXPECTED OUTPUT

9

OUTPUT

9

OPTIONS

Editor ☒ File Upload ☐

Compare with Expected Output:
☒ Yes ☐ No

Language :

Compile With Input : ☒ yes ☐ No

FILE UPLOADER

Upload main file :

Upload input file :

COMPILE & EXECUTE

RESULT

Actual output : 9
Expected output : 9
Status : PASS

Single File-Multiple source codes File Upload

C-JUDGE
C/C++ PROGRAM EVALUATOR

HOMEEMULTIPLE TEST CASESMULTIPLE-FILE PROGRAMMULTIPLE SOURCE PROGRAMS

CODE UPLOAD

Upload multiple main files :

INPUT

Upload input file :

EXPECTED OUTPUT

Upload expected output file :

OPTIONS

Language :

Compile With Input : ☒ yes ☐ No

Compare with Expected Output:
☒ yes ☐ No

COMPILE & EXECUTE

STATISTICS

PASS/FAIL STATISTICS

Category	Percentage
PASS	66.7%
FAIL	33.3%

RESULT

Results :

Name	Output	Expected Output	Status
sum2.cpp	8	11	FAIL
sum1.cpp	11	11	PASS
sum3.cpp	11	11	PASS

Multiple Files-Single source code File Upload

C-JUDGE
C/C++ PROGRAM EVALUATOR

[HOME](#) [MULTIPLE TEST CASES](#) **[MULTIPLE-FILE PROGRAM](#)** [MULTIPLE SOURCE PROGRAMS](#)

CODE UPLOAD

Upload source code file : [Browse...](#)

INPUT

Upload input file : [Browse...](#)

EXPECTED OUTPUT

Upload expected output file : [Browse...](#)

OPTIONS

Language :

Compile With Input : ☒ yes ☐ No

Compare with Expected Output:
☒ yes ☐ No

COMPILE & EXECUTE

[Compile&Execute](#)

RESULT

Results :

Output	Expected Output	Status
1 8 9 10 70	1 8 9 10 70	PASS

Multiple Test Cases-Single source code File Upload

C-JUDGE
C/C++ PROGRAM EVALUATOR

HOMEMULTIPLE TEST CASESMULTIPLE-FILE PROGRAMMULTIPLE SOURCE PROGRAMS

CODE EDITOR

```
#include<stdio.h>

int main()
{
    int a , b, c ;
    scanf("%d",&a);
    scanf("%d",&b);
    c = a + b ;
    printf("%d",c);
}
```

OPTIONS

Language : C

Compile With Input : ☒ yes ☐ No

FILE UPLOADER

Editor : ☒

File Upload : ☐


Upload Source Code :
Choose Files No file chosen

COMPILE & EXECUTE

Compile & Execute

STATISTICS

PASS/FAIL STATISTICS



PASS
FAIL

UPLOAD MULTIPLE TEST CASES

INPUT

Choose File No file chosen

EXPECTED OUTPUT

Choose File No file chosen

SUBMIT TEST CASES

Submit Files

RESULT

Results :

TestCase	Input	Output	Expected Output	Status
Test_case3	3 5	8	11	FAIL
Test_case2	4 6	10	10	PASS
Test_case1	3 5	8	10	FAIL

Design Challenges

- Understanding and utilizing the components of Express Framework
- Need of different components for each feature
- Information exchanges between components
- Handling large number of inputs – storage, processing etc.
- Program must produce output in predefined format. Noncompliance will result in failed test.

Future Work

- Integration of plagiarism detector
- Authentication
- Progress tracking for evaluation

Thank you