

WebAssemblyを用いた ブラウザ上で完結する C言語のTDD式学習システムの構築と提案

1. はじめに
2. 研究方法と研究意義
3. DEMO
4. 実装方法と構築したシステムについて
5. 研究の評価について
6. まとめ

1. はじめに

高専・大学における工学系

→ C言語が必要言語の1つ

呉高専でも5年間プログラムを書く機会がある

しかし学生によって習得度合いの開きが感じられた

1. はじめに

要因は何か？

- ・ 授業方法や実習方法に問題がある ➡ 方法の問題
- ・ 授業時間や実習時間が少ない ➡ 時間の問題
- ・ 授業時間外でする機会が少ない ➡ 機会の問題
- ・ コードを写すだけで理解していない ➡ 意欲, 方法の問題



授業や課題で,

“自分が何をしているか, 何を作っているのか想像がつかない”
という問題が生まれてしまっている

限られた授業時間内で
効率的 & 効果的
に学習できるシステムがあれば…

ブラウザ上で完結し、
軽量かつテスト可能な
学習環境を提案する

2. 研究方法と研究意義

“ブラウザ上で完結する”

→ 授業時間外でプログラミングをする機会が少ない(機会の問題)

自宅や寮で自習をするには、PCに授業と同じ学習環境を用意する必要がある。しかし学習環境を用意するのは困難。

今回構築したシステムはブラウザ上で完結している

→ 学習者が用意するものはブラウザのみ

→ プログラミングを始める壁を低くできる

→ 自主学習を促すことへ繋がる

2. 研究方法と研究意義

“軽量”

→ 授業時間外でプログラミングをする機会が少ない(機会の問題)

学習環境の一つに、統合開発環境(IDE)も挙げられるが、一般に推奨環境のスペックが高いため、

→ 学校などの既存PCから置き換えること

→ 学生に対し高額なPCの購入を求めること

は難しい

ブラウザ上で動作する

→ 安価で性能の低いPCでも十分動作する

→ 現在の環境からへの移行が容易かつ素早く行える

2. 研究方法と研究意義

“テスト可能”

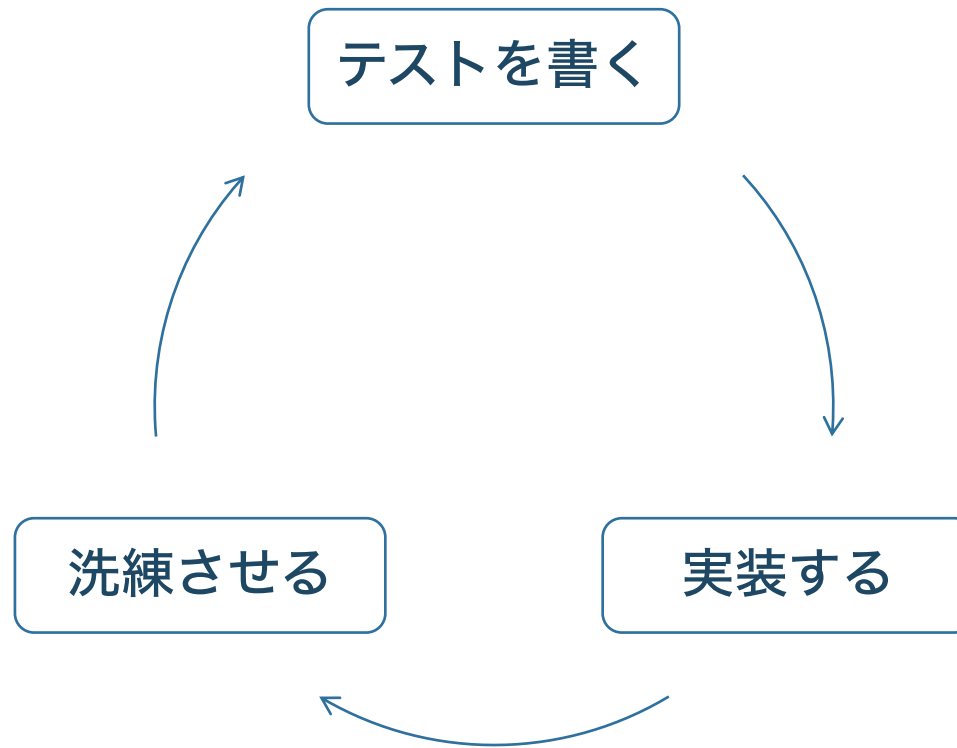
- 授業方法や実習方法に問題がある(方法の問題)
- 授業時間や実習時間が少ない(時間の問題)
- コードを写すだけで理解していない(意欲, 方法の問題)

用語説明：テストとは

テスト≡テスト駆動開発(Test-Driven Development)

プログラム開発手法の一種

“プログラムが正しいかどうかを機械的に判断するもの”



引数のうち最も大きい値を返す関数

```
1 int max(int a, int b, int c) {  
2     if (a > b && a > c) return a;  
3     if (b > a && b > c) return b;  
4     if (c > a && c > b) return c;  
5 }  
6  
7 assert(max(1, 2, 3) == 3); // Pass  
8 assert(max(1, 1, 1) == 1); // Fail
```

上記コードは間違っているためテストを通らない

2. 研究方法と研究意義

“テスト可能”

- 授業時間や実習時間が少ない(時間の問題)
- コードを写すだけで理解していない(意欲, 方法の問題)

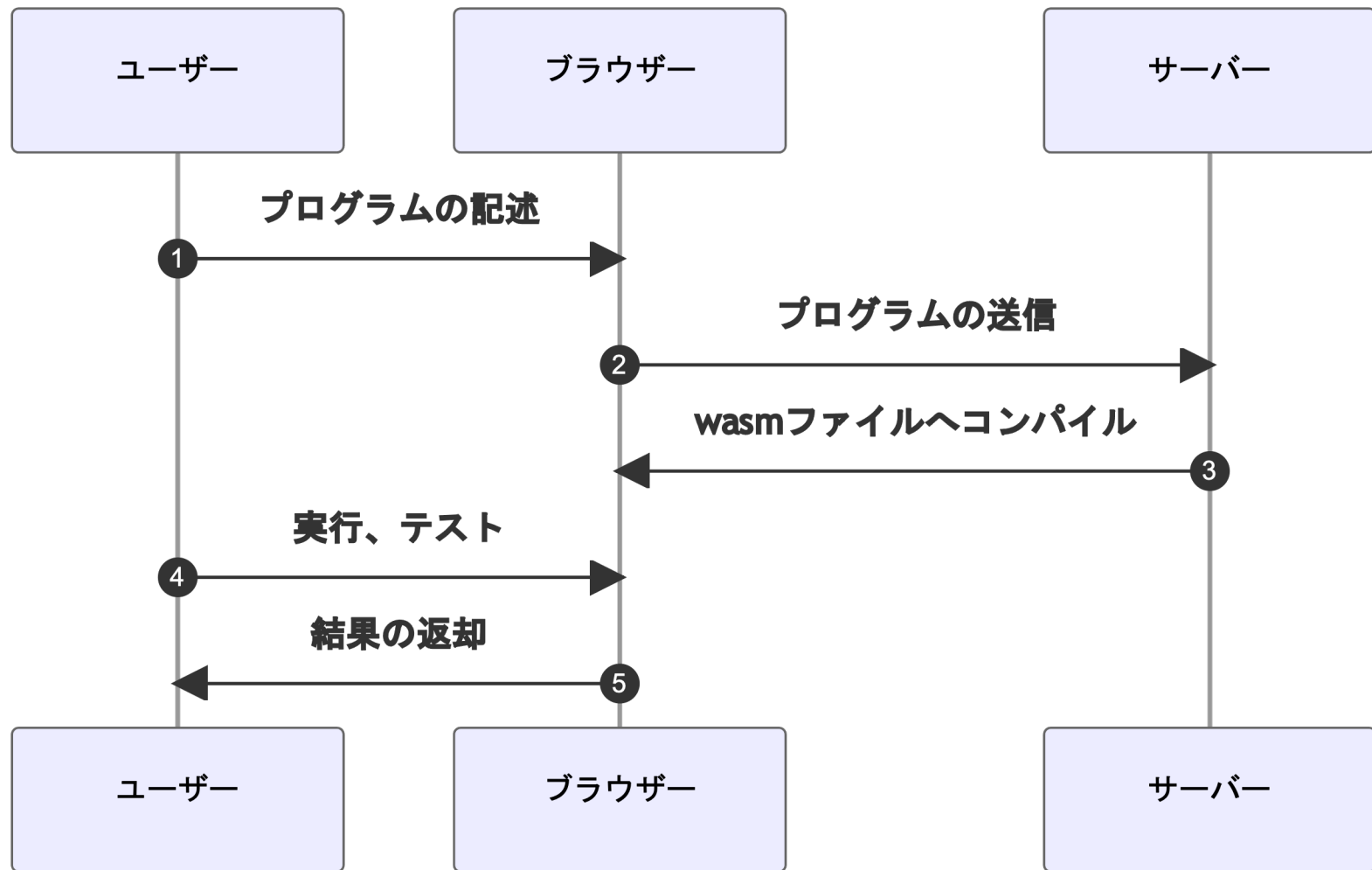
学習者自身が“課題を達するのに必要な知識や問題の細分化”
を行えていない可能性

テストがあると

- 標準入力による動作確認の簡略化, 時間短縮
 - トライアンドエラーの高速化, 効率化
- テストケースからプログラムの具体的なイメージがつかめる
 - 完成や動作を意識した学習になる

3. DEMO

4. 実装方法と構築したシステムについて



4. 実装方法と構築したシステムについて

コンパイル

実行

標準入出力

テスト

整形

ダウンロード

補完機能

自動保存

ツールチップヒント

クイックフィックス

URLによる共有

環境変数

コマンドライン引数

4. 実装方法と構築したシステムについて

整形

ルールに従ってプログラムの動作を変えずに見た目を整える機能.

ツールチップヒント

変数や関数にマウスをホバーするとその説明や詳細が表示される機能.

クイックフィックス

機械的に修正可能なものを修正する機能（例：セミコロン忘れなど）.

補完機能

宣言済みの変数やfor文などの構文が入力時に補完される機能.

自動保存

500ms 毎にブラウザにプログラム等を保存. ブラウザを閉じても再度復元可能.

URLによる共有

プログラムなどを符号化しURLに付与する. ブラウザで開くだけで同じ状態を再現できる.

4. 実装方法と構築したシステムについて

“単体テストを可能にするために”

テストを行うには関数に直接引数を渡す必要がある

WebAssemblyでは数値のみしか扱えない

→ 全ての引数を数値として扱わなければならない

→ 文字列はどうか

WebAssemblyは線形メモリを持っている

→ 文字列長と文字列をエンコードしたバイトを格納

→ 単体テストには開始アドレスを渡す

4. 実装方法と構築したシステムについて

Console

Memory Inspector x

Issues

What's New

Network request blocking

Memory(1) x

< 0x00000000 >

00000000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

< 0x00000000 >

00000000 03 00 00 00 61 62 63 00 00 00 00 00 00 00 00 00

00000010 18 00 00 00 00 04 00 00 00 00 00 00 00 00 00 00

00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

< 0x00000000 >

00000000 03 00 00 00 21 62 63 00 00 00 00 00 00 00 00 00

00000010 18 00 00 00 00 04 00 00 00 00 00 00 00 00 00 00

00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Little Endian ▾

Integer 8-bit dec ▼ 0

Integer 16-bit dec ▼ 0

Integer 32-bit dec ▼ 0

Integer 64-bit dec ▼ 0

Float 32-bit dec ▼ 0.00

Float 64-bit dec ▼ 0.00

Pointer 32-bit nvn ↻

Little Endian ▾

Integer 8-bit dec ▼ 3

Integer 16-bit dec ▼ 3

Integer 32-bit dec ▼ 3

Integer 64-bit dec ▼ 27974191445704707

Float 32-bit dec ▼ 0.00

Little Endian ▾

Integer 8-bit dec ▼ 3

Integer 16-bit dec ▼ 3

Integer 32-bit dec ▼ 3

Integer 64-bit dec ▼ 27973916567797763

Float 32-bit dec ▼ 0.00

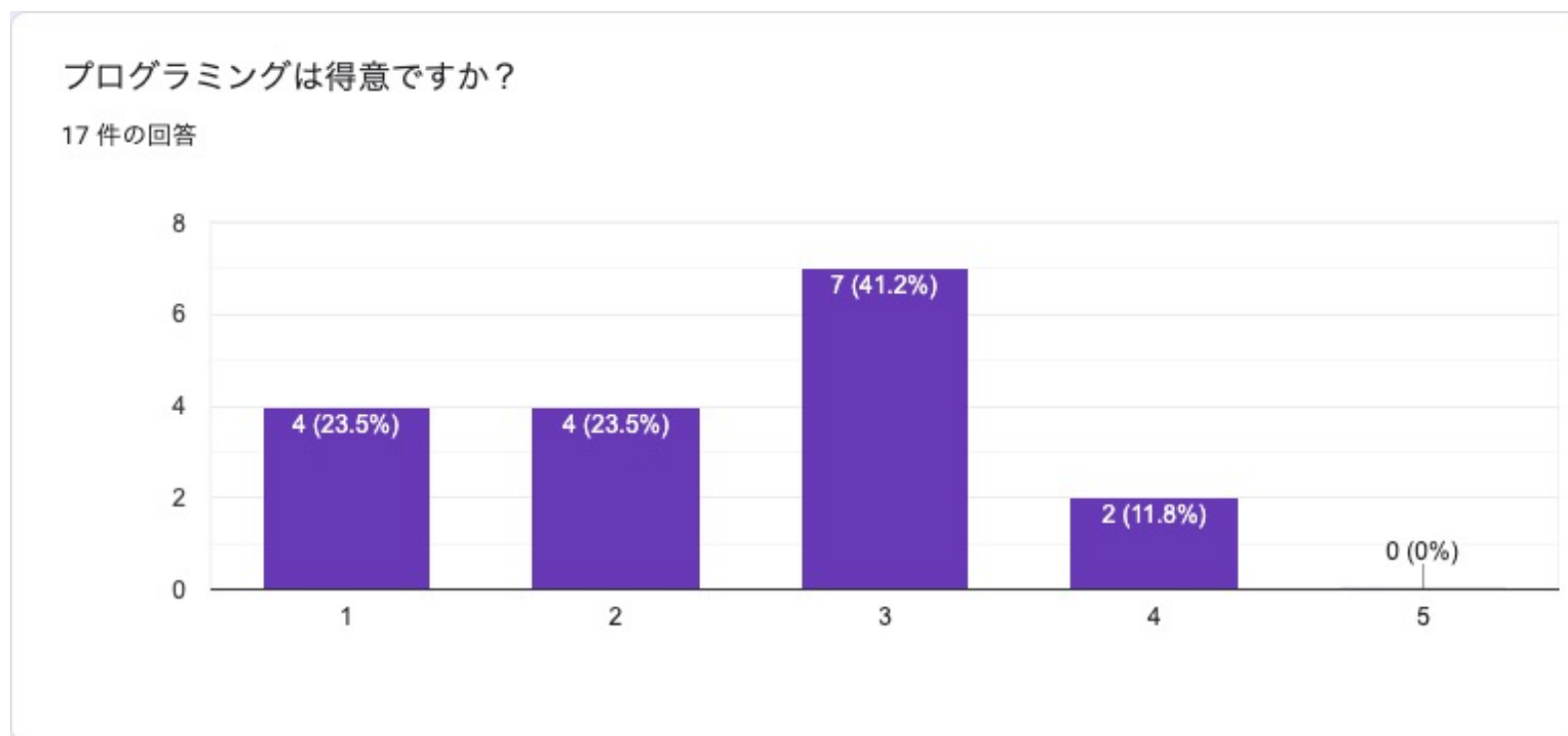
Float 64-bit dec ▼ 0.00

Pointer 32-bit 0x3 ↻

Pointer 64-bit 0x63622100000003 ↻

5. 研究の評価について

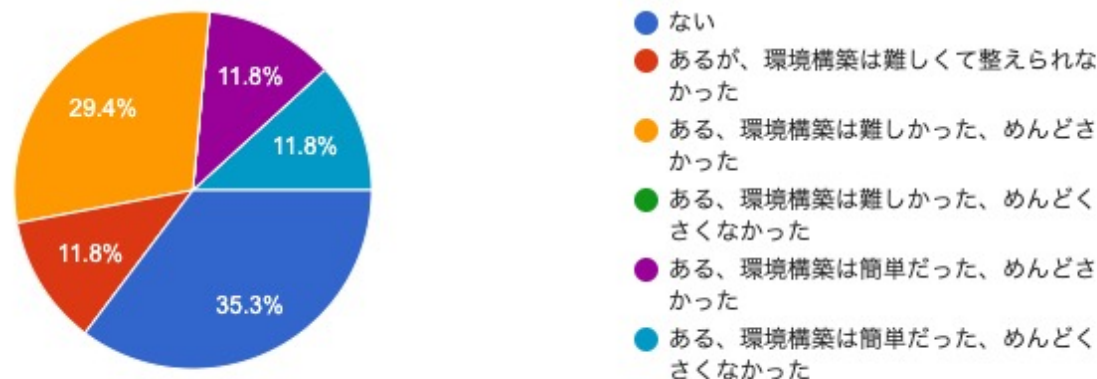
本研究では構築したシステムが「はじめに」に記した通りの効果を得られるかどうかの評価に、呉工業高等専門学校 電気情報工学科に在学する学生に対し、用意したテストを満たすように関数を実装してもらい、その利便性や使い勝手などに関するアンケート調査を17名の学生を対象に行なった。



5. 研究の評価について

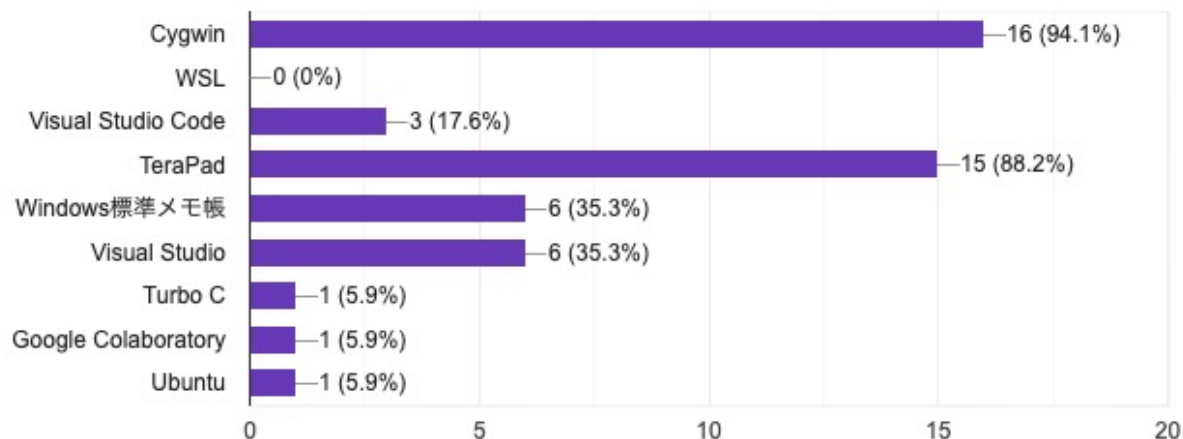
自分のPCに予習や復習のために前述のような学校と同じ学習環境を整えたことはありますか

17 件の回答



授業の学習に使った環境はなんですか？

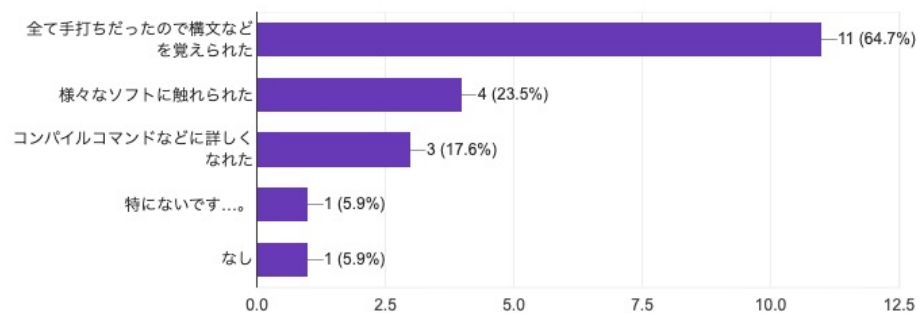
17 件の回答



5. 研究の評価について

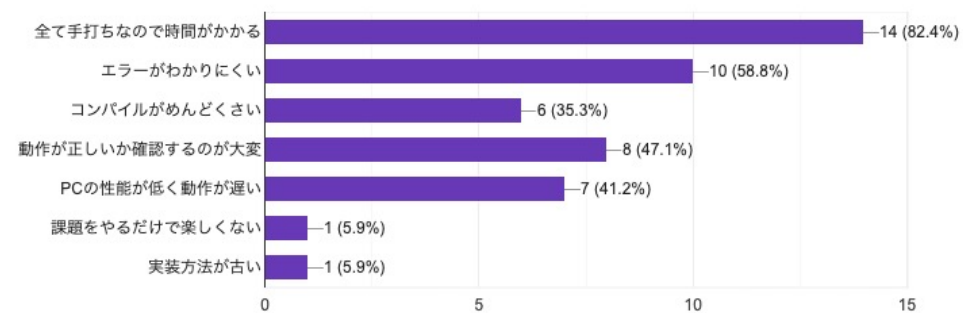
授業の良かったところを教えてください

17 件の回答



授業の良くなかったところを教えてください

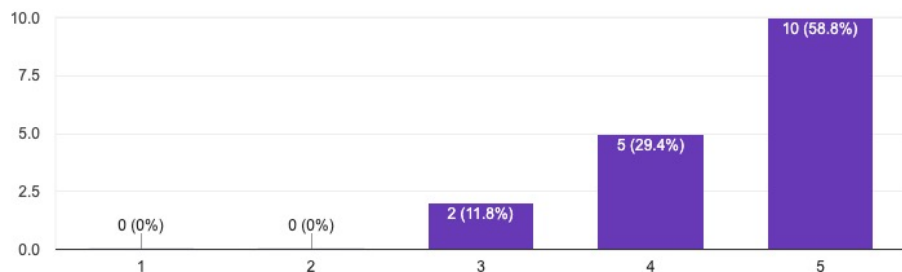
17 件の回答



5. 研究の評価について

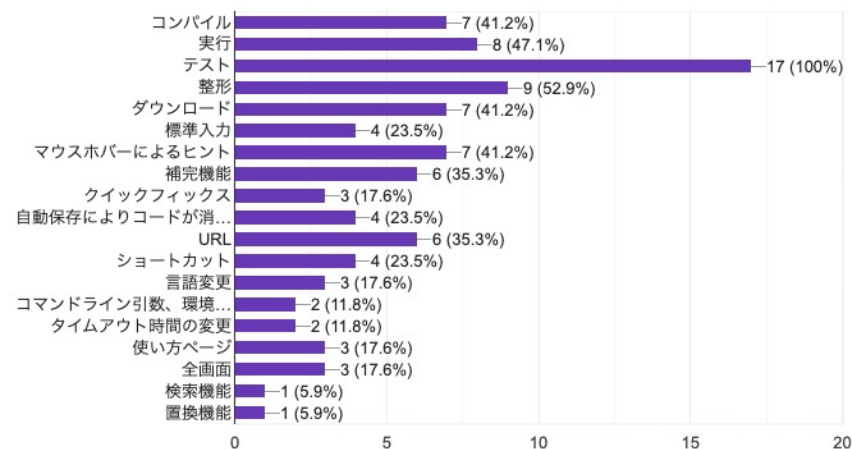
動作速度について教えてください

17 件の回答



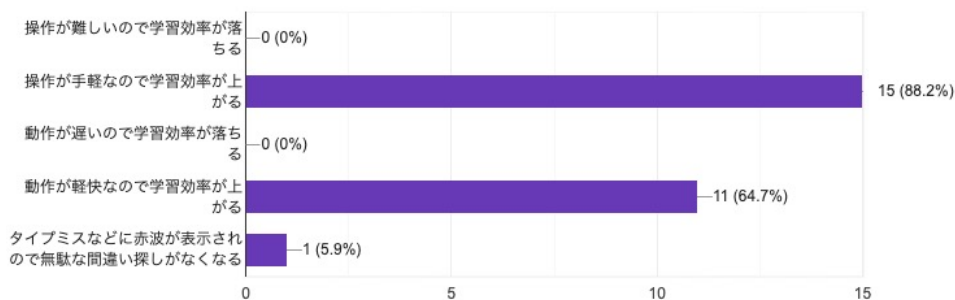
良かった点について教えてください

17 件の回答



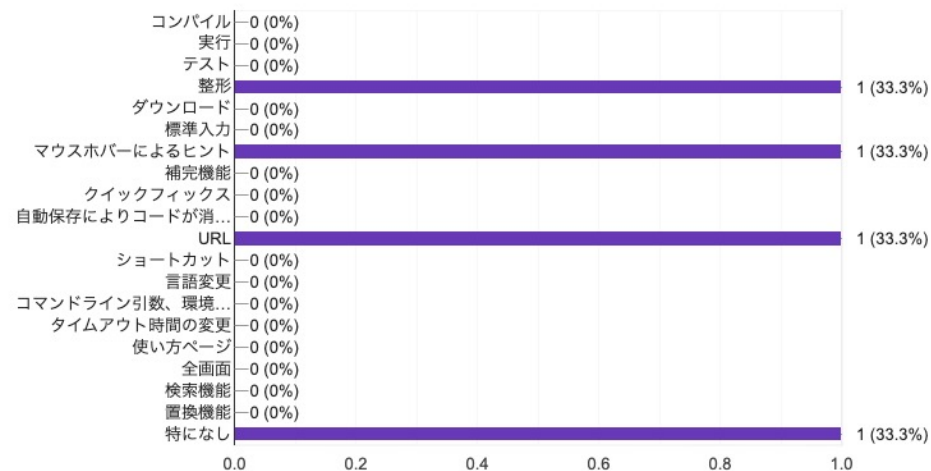
授業で現在の学習方法から本システムに変更された場合の影響について教えてください

17 件の回答



悪かった点について教えてください

3 件の回答



5. 研究の評価について

自由記入

”テストにより失敗したときに何ができて何ができないのかが場合分けされていたので間違いを分析しやすい”

”自宅と学校を同じ環境にするのは大変だが、これなら簡単に使用できるためわかりやすい”

という意見があった。

6. まとめ

わかったこと

- 多くの学生が現在の授業方法に問題を感じている
- 構築したシステムでその問題は大きく改善されること

実際の学習効果について

- 授業などで使い，使用状況や使い勝手などを調査する

今後

- 今回のアンケートで得られた不満点などの修正
- コンパイルエラーを日本語にする
 - 多くの学生がコンパイルエラーをきちんと読むことにつながる
 - コンパイラのエラー類がハードコーディングされている問題を解決することが直近の課題