ad-2. 双方向リスト

アルゴリズムとデータ構造

（スライド，全１１回）

https://www.kkaneko.jp/cc/ae/index.html

金子邦彦

双方向リストとは

レコードを次の２つで構成

- 要素を格納する**セル**
- **リスト**中の**次**の**レコード**を指す**ポインタ**を格納する**セル**
- **リスト**中の**前**の**レコード**を指す**ポインタ**を格納する**セル**

双方向リストと連結リストの違い

リスト内のある**レコード**のアドレス A が分かっているとする.

- **連結リスト**

A の**次**に要素を挿入することは簡単.

A の**前**に要素を挿入することはできない.

- **双方向リスト**

A の**次**に要素を挿入することは簡単.

A の**前**に要素を挿入することも簡単.

1-2. 実習

実習の指示

- 資料：**６～１２**
- 次のことを理解しマスターする
  - 双方向リスト

実習

① **ウェブブラウザ**を起動する

② **C Tutor** を使いたいので，次の URL を開く

http://www.pythontutor.com/

※ **Internet Explorer でうまく動かない**場合がある

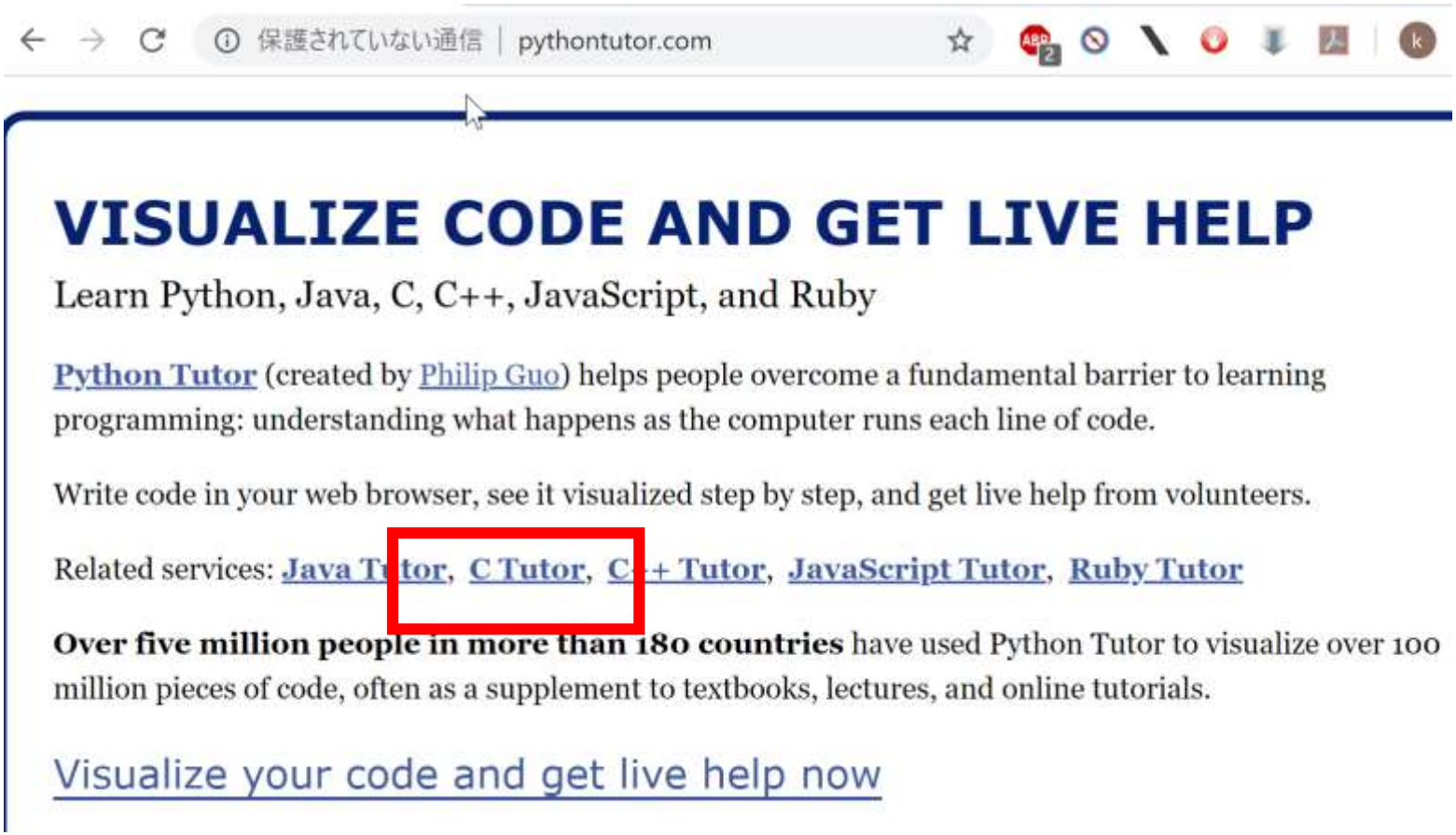
→ うまく動かないときは **Google Chrome** を試してください

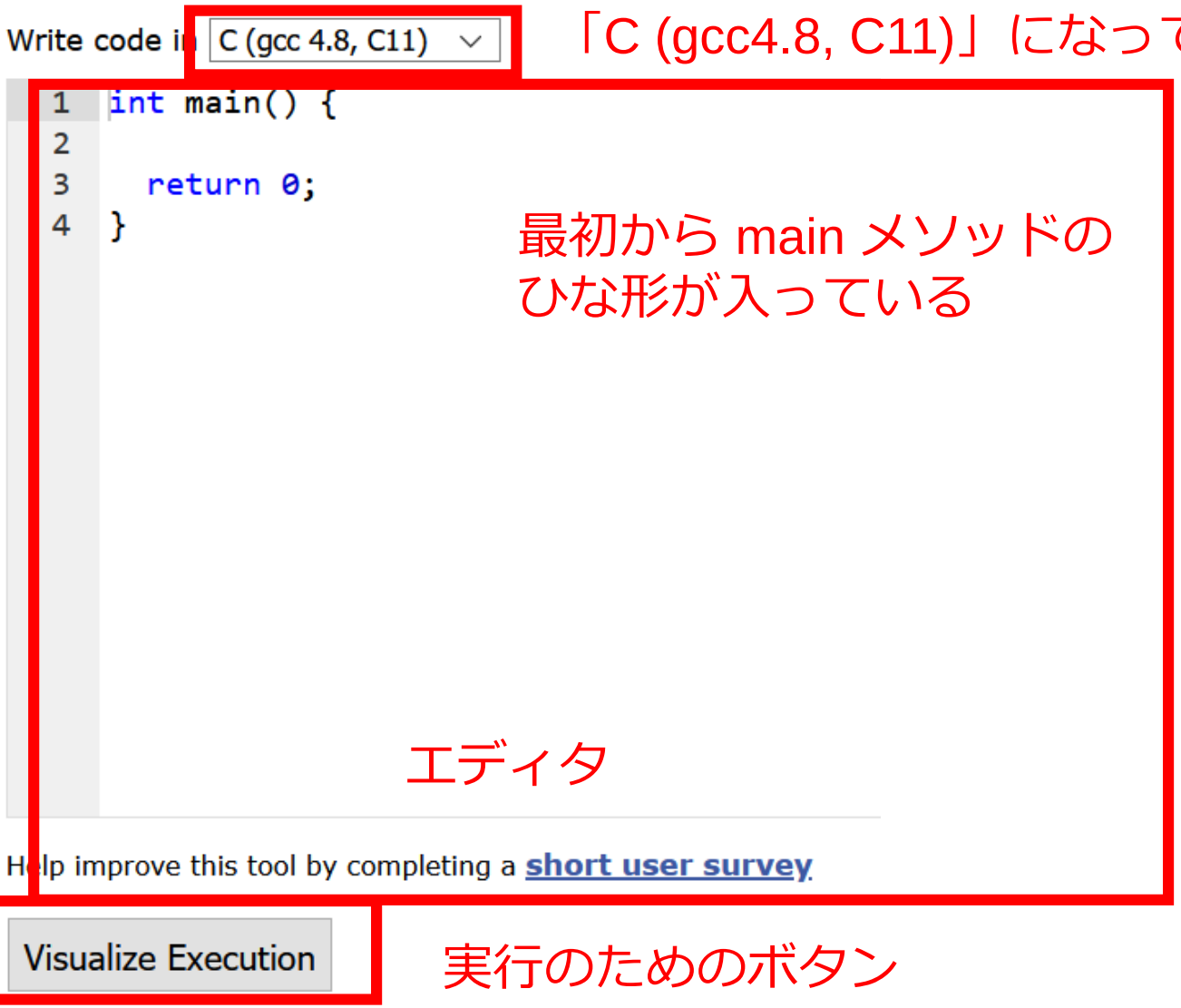
※ 途中で**「Server Busy・・・」というメッセージが出る**ことがある.

→ 混雑している．**少し（数秒から数十秒）待つ**と自動で表示が変わる（変わらない場合には，操作をもう一度行ってみる）

※ 日本語モードはない．英語で使う

③ 「C Tutor」をクリック




Write code in
C (gcc 4.8, C11)
「C (gcc4.8, C11)」になっている
1 int main() {
2
3 return 0;
4 }
最初から main メソッドの
ひな形が入っている
エディタ
Help improve this tool by completing a short user survey
Visualize Execution
実行のためのボタン

双方向リストの作成

④ 次のプログラムを使う

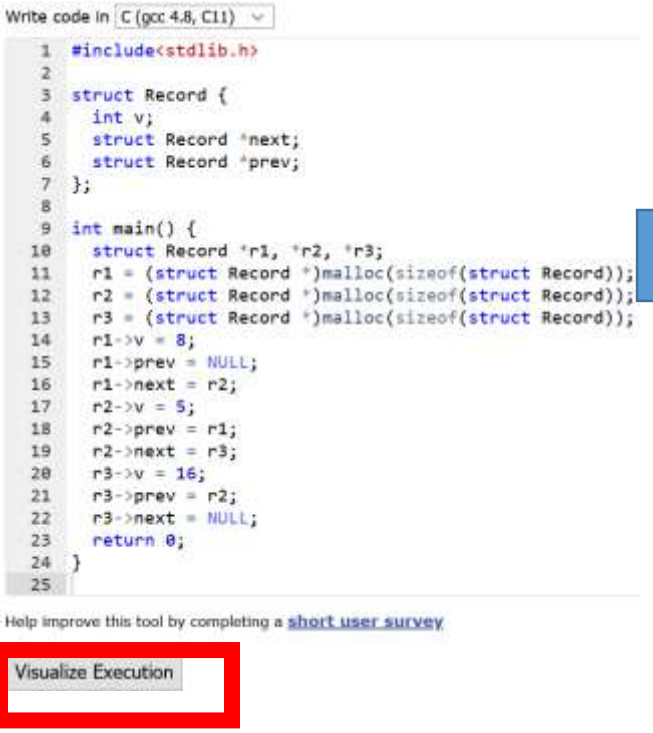
```c
#include<stdlib.h>

struct Record {
  int v;
  struct Record *next;
  struct Record *prev;
};

int main() {
  struct Record *r1, *r2, *r3;
  r1 = (struct Record *)malloc(sizeof(struct Record));
  r2 = (struct Record *)malloc(sizeof(struct Record));
  r3 = (struct Record *)malloc(sizeof(struct Record));
  r1->v = 8;
  r1->prev = NULL;
  r1->next = r2;
  r2->v = 5;
  r2->prev = r1;
  r2->next = r3;
  r3->v = 16;
  r3->prev = r2;
  r3->next = NULL;
  return 0;
}
```

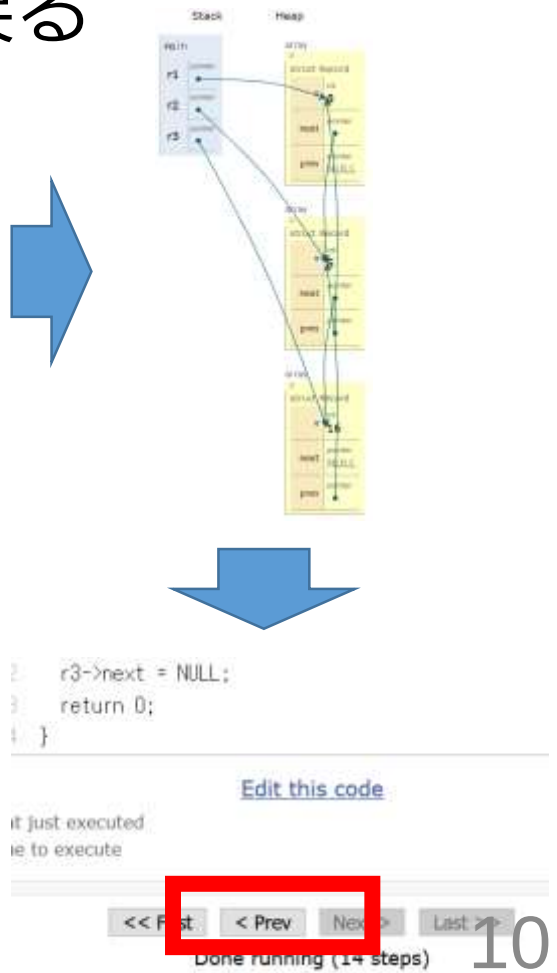
②「**Visualize Execution**」をクリック.

「**Last**」をクリック.

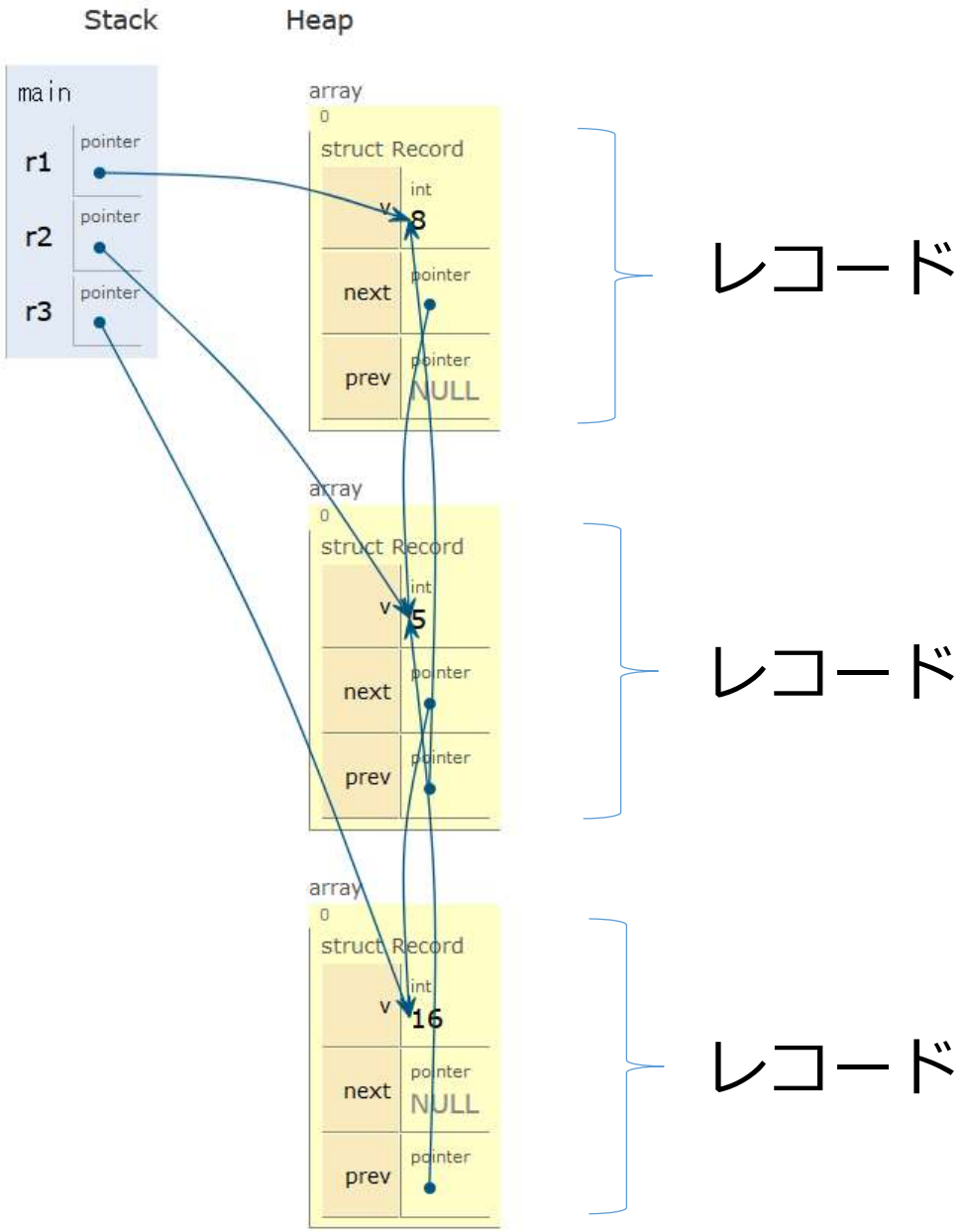
結果を確認する.

「**Edit this code**」をクリックして戻る





実行結果


Stack
Heap
main
r1
pointer
r2
pointer
r3
pointer
array
0
struct Record
v
int
8
next
pointer
prev
pointer
NULL
array
0
struct Record
v
int
5
next
pointer
prev
pointer
array
0
struct Record
v
int
16
next
pointer
NULL
prev
pointer
レコード
レコード
レコード

実習

- いまのプログラムで

  末尾の要素（値は16）を削除するプログラム

  を書き加えなさい

・いまのプログラムで，

  末尾に新しい要素（値は 24）を挿入する

  プログラムを書き加えなさい