

プログラミングII

<http://bit.ly/Prog3i>

言語処理系 (3)

前期 第12週

2019/7/10

本日は・・・

前回までの電卓とは別の字句解析・構文解析のプログラムを作ってみます。

本日は・・・

**【例】 整数の配列に対して操作をする
コマンドを実行するプログラム**

- ▶ **配列の初期化**
- ▶ **配列の値の出力**
- ▶ **配列の要素に値を代入する**
- ▶ **配列の全要素の合計を求める**
- ▶ **などなど**

【課題の準備】

演習室で作業する前に、以下のコマンドを
入れるだけで準備が完了する

```
$ mygitclone 「自分のGitHubユーザ名」  
$ cd prog3i-(ユーザ名)  
$ ./myconf
```

※本体をシャットダウンするまでは、
上記「mygitclone」と「myconf」の設定は有効です

【課題の準備】

以下の流れで、課題のプログラムを作るためのフォルダを準備しましょう。

1. 端末を起動して、以下のコマンドを実行して前期第12週のフォルダを作る
\$ cd prog3i-(ユーザ名) (←既に移動しているなら不要)
\$ mkdir week112
\$ cd week112

【練習12-1】

サンプルプログラムを使って、
「配列の初期化（clear）」と「プログラムの終了
（exit）」の動作を確認して下さい。

【課題12-1】

「配列の全要素の値を出力する」 コマンドshowを追加して下さい。

- ▶ array.lexに、コマンド「show」と識別子SHOWの字句定義を追記する
- ▶ array.yaccに、このコマンドに関する構文を追加する
EXIT, CLEARと同様にコマンドの後には何も続かない
- ▶ myproc.hにこの演算で呼び出す関数のプロトタイプ宣言を追加する
`int show();`
- ▶ myproc.cに関数showの定義を追加する
配列aの0番目～9番目の値を出力する
最後に0を戻す

【課題12-1】

[実行結果]

```
$ ./a.out
```

```
clear
```

```
show
```

```
0 0 0 0 0 0 0 0 0 0
```

```
exit
```


【課題の提出】

以下の流れで、作ったCプログラムをGitHubにプッシュして、Webサイトで確認してみましょう。

1. 端末内で、以下のコマンドで課題を提出

```
$ git add -A
```

```
$ git commit -m “課題12-1提出”
```

```
$ git push origin master
```

2. 自分のリポジトリを開いて、提出したファイルがプッシュされているか確認する

[https://github.com/nit-ibaraki-prog3i/prog3i-\(ユーザ名\)](https://github.com/nit-ibaraki-prog3i/prog3i-(ユーザ名))

【課題12-2】

「配列の*i*番目に値*n*を代入する」 コマンドsetを追加して下さい。

- ▶ array.lexに、コマンド「set」と識別子SETの字句定義を追記する
- ▶ array.yaccに、このコマンドに関する構文を追加する
コマンドの後に、NUMBERが2個続く
- ▶ myproc.hにこの演算で呼び出す関数のプロトタイプ宣言を追加する

```
int set(int i, int n);
```
- ▶ myproc.cに関数setの定義を追加する
配列aの*i*番目に*n*を代入する
最後に0を返す

【課題1 2-2】

[実行結果]

```
$ ./a.out
```

```
clear
```

```
set 3 8
```

```
set 7 2
```

```
show
```

```
0 0 0 8 0 0 0 2 0 0
```

```
exit
```

【課題の提出】

以下の流れで、作ったCプログラムをGitHubにプッシュして、Webサイトで確認してみましょう。

1. 端末内で、以下のコマンドで課題を提出

```
$ git add -A
```

```
$ git commit -m “課題12-2提出”
```

```
$ git push origin master
```

2. 自分のリポジトリを開いて、提出したファイルがプッシュされているか確認する

[https://github.com/nit-ibaraki-prog3i/prog3i-\(ユーザ名\)](https://github.com/nit-ibaraki-prog3i/prog3i-(ユーザ名))

【課題12-3】

「配列の全要素の合計を出力する」コマンドsumを追加して下さい。

- ▶ array.lexに、コマンド「sum」と識別子SUMの字句定義を追記する
- ▶ array.yaccに、このコマンドに関する構文を追加する
コマンドの後には何も続かない
- ▶ myproc.hにこの演算で呼び出す関数のプロトタイプ宣言を追加する
`int sum();`
- ▶ myproc.cに関数sumの定義を追加する
配列aの0番目～9番目の値の合計を求めて出力する
最後に0を戻す

【課題1 2-3】

[実行結果]

```
$ ./a.out
```

```
clear
```

```
set 0 6
```

```
set 9 8
```

```
show
```

```
6 0 0 0 0 0 0 0 0 0 8
```

```
sum
```

```
14
```

```
exit
```

【課題の提出】

以下の流れで、作ったCプログラムをGitHubにプッシュして、Webサイトで確認してみましょう。

1. 端末内で、以下のコマンドで課題を提出

```
$ git add -A
```

```
$ git commit -m “課題12-3提出”
```

```
$ git push origin master
```

2. 自分のリポジトリを開いて、提出したファイルがプッシュされているか確認する

[https://github.com/nit-ibaraki-prog3i/prog3i-\(ユーザ名\)](https://github.com/nit-ibaraki-prog3i/prog3i-(ユーザ名))

【課題12-4】

「s番目～e番目の要素に値nを代入する」 コマンド
set2を追加して下さい。 ($s < e$ を前提としてよい)

- ▶ array.lexに、コマンド「set2」と識別子SET2の字句定義を追記する
- ▶ array.yaccに、このコマンドに関する構文を追加する
コマンドの後に、NUMBERが3個続く
- ▶ myproc.hにこの演算で呼び出す関数のプロトタイプ宣言を追加する
`int set2(int s, int e, int n);`
- ▶ myproc.cに関数set2の定義を追加する
配列aのs番目～e番目に値nを代入する
最後に0を戻す

【課題12-4】

[実行結果]

```
$ ./a.out  
clear
```

```
set2 2 7 5
```

```
show
```

```
0 0 5 5 5 5 5 5 0 0
```

```
set2 6 9 1
```

```
show
```

```
0 0 5 5 5 5 1 1 1 1
```

```
exit
```

【課題の提出】

以下の流れで、作ったCプログラムをGitHubにプッシュして、Webサイトで確認してみましょう。

1. 端末内で、以下のコマンドで課題を提出

```
$ git add -A
```

```
$ git commit -m “課題12-4提出”
```

```
$ git push origin master
```

2. 自分のリポジトリを開いて、提出したファイルがプッシュされているか確認する

[https://github.com/nit-ibaraki-prog3i/prog3i-\(ユーザ名\)](https://github.com/nit-ibaraki-prog3i/prog3i-(ユーザ名))

まだ余裕のある人は…【課題12-5】

「配列の全要素の平均を出力する」コマンドaveを追加して下さい。

- ▶ array.lexに、コマンド「ave」と識別子AVEの字句定義を追記する
- ▶ array.yaccに、このコマンドに関する構文を追加する
コマンドの後には何も続かない
- ▶ myproc.hにこの演算で呼び出す関数のプロトタイプ宣言を追加する
`int ave();`
- ▶ myproc.cに関数aveの定義を追加する
配列aの0番目～9番目の値の平均を求めて出力する
最後に0を戻す

【課題12-5】

[実行結果]

```
$ ./a.out  
clear
```

```
set2 0 5 3
```

```
set2 6 9 7
```

```
show
```

```
3 3 3 3 3 3 7 7 7 7
```

```
ave
```

```
4.600000
```

```
exit
```

【課題の提出】

以下の流れで、作ったCプログラムをGitHubにプッシュして、Webサイトで確認してみましょう。

1. 端末内で、以下のコマンドで課題を提出

```
$ git add -A
```

```
$ git commit -m “課題12-5提出”
```

```
$ git push origin master
```

2. 自分のリポジトリを開いて、提出したファイルがプッシュされているか確認する

[https://github.com/nit-ibaraki-prog3i/prog3i-\(ユーザ名\)](https://github.com/nit-ibaraki-prog3i/prog3i-(ユーザ名))

小テストについて

小テストの注意点

- 他人の力は借りずに、自分だけでプログラムを作成する。（つまり定期試験と同様）
- プログラムの提出はGitHubを使用する。

小テストについて

小テスト中に参照できるもの

- 教科書, 参考書, 配付資料
- 自分のホームディレクトリ（ホームフォルダ）以下に保存されているファイル
- 小テストでは紙媒体のものは参照可能
- 上記以外の情報を参照することは不正行為とする
例：USBで接続された機器に保存されているファイルの参照
Webブラウザ、ネットワークを介した情報の参照
自分のPCを使用する、など