

[演習L] 字句解析の演習

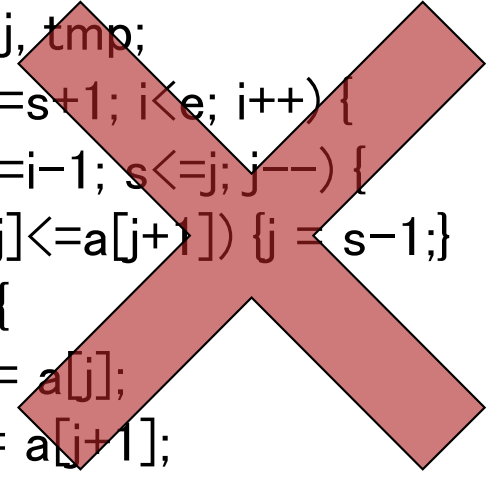
- 資料をダウンロード (&プリント)
 - ❖ 講義WWW→講義ノート→4. 字句解析の演習
- 資料に従ってプログラム類をDL
 - ❖ プロⅢやLUNA等と同じ認証
- 提出はWWW
 - ❖ 講義WWW→レポート提出
 - ❖ **締切を確認すること【超重要】**
 - ❖ アカウントのない人⇒石浦まで

必ずきちんと動かすこと

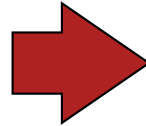
- 構文解析 (コンパイラ作成) 演習で使う
- 指示通り diff で動作確認すること
- 不完全動作は減点&再提出
- 行き詰まったら、石浦/石浦研究室へ
 - ❖ メール
 - ◆ プログラムをレポート提出にアップロード
 - ◆ 何をしたらどうなったか状況説明
 - ❖ 直接
 - ◆ USBメモリーでプログラム持参

必ずインデントすること

```
void si_sort(int a[ ], int s, int e)
{
  int i, j, tmp;
  for (i=s+1; i<e; i++) {
  for (j=i-1; s<=j; j--) {
  if (a[j]<=a[j+1]) {j = s-1;}
  else {
  tmp = a[j];
  a[j] = a[j+1];
  a[j+1] = tmp;
  }}}}
```



構造がわからない
(思考できない)



```
void si_sort(int a[ ], int s, int e)
{
  int i, j, tmp;
  for (i=s+1; i<e; i++) {
    for (j=i-1; s<=j; j--) {
      if (a[j]<=a[j+1]) {j = s-1;}
      else {
        tmp = a[j];
        a[j] = a[j+1];
        a[j+1] = tmp;
      }
    }
  }
}
```

インデントの流儀

BSD/オールマン スタイル

```
for (i=0; i<N; i++)  
{  
    if (i % 2 == 0)  
    {  
        s = s + i;  
    }  
}
```

VS

K&Rスタイル

```
for (i=0; i<N; i++) {  
    if (i % 2 == 0) {  
        s = s + i;  
    }  
}
```

- ① 行数少ない
- ② 行末の ; 防止

```
if (i % 2 == 0);  
{  
    s = s + i;  
}
```

どちらでもよい (好みで)

場合分けは if～else if～…else

もし A なら ⇒ 処理 X
そうでなくて B なら ⇒ 処理 Y
そうでなくて C なら ⇒ 処理 Z
そうでないなら ⇒ 処理 U



```
if ( A ) { 処理 X }  
else if ( B ) { 処理 Y }  
else if ( C ) { 処理 Z }  
else { 処理 U }
```

```
if ( A ) { 処理 X }  
if ( B ) { 処理 Y }  
if ( C ) { 処理 Z }  
else { 処理 U }
```

コーディング規約/ガイドライン

□ 文法規則以外のルール

- ❖ 例) #define の変数は大文字
- ❖ 読み易くするため/間違いを防ぐため
- ❖ 組織毎に取り決められていることが多い

□ MISRA-C

- ❖ MISRA: Motor Industry Software Reliability Association
- ❖ 組み込みSW用Cのガイドライン
- ❖ C90準拠 + 141項目のルール
- ※ 組み込みSWに特化し過ぎたルールもある

最後の else 節は省略しない

```
if ( D==0 ) { num_sol = 1; }  
else if ( D>0 ) { num_sol = 2; }  
else if ( D<0 ) { num_sol = 0; }
```



```
if ( D==0 ) { num_sol = 1; }  
else if ( D>0 ) { num_sol = 2; }  
else { num_sol = 0; }
```

```
if ( D==0 ) { num_sol = 1; }  
else if ( D>0 ) { num_sol = 2; }  
else if ( D<0 ) { num_sol = 0; }  
else { ; }
```

MISRA-C Rule 14.10

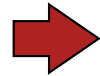
if ~ else if ~ は必ず
else で終わること

あるいは
else { assert(0); }

if, for, while の { }

本体が 1 文でも { } をつけること

```
for (i=0; i<N; i++)  
    if (i % 2 == 0) s = s + i;
```



```
for (i=0; i<N; i++) {  
    if (i % 2 == 0) { s = s + i; }  
}
```

理由① インデントのミス
が致命的エラーに

MISRA-C Rule 14.8, 14.9

```
for (j=i-1; s<=j; j--) {  
    if (a[j]<=a[j+1])  
        j = s-1;  
    tmp = a[j];  
    a[j] = a[j+1];  
    a[j+1] = tmp;  
}
```

理由② マクロの複文でミスし易い

```
#define PUSH(x) sp++; stack[sp]=x  
  
if (sp<MAX) PUSH(x);
```