

状態遷移の考え方をプログラムに応用する。

まず、次のような関数を作成してみよう。

関数名 : `buffer_to_array2D`

引数 : `char buffer[200]` , `char note[10][20]`

戻り値 : なし

副作用 : `buffer` 中の英単語を `note` に格納する。スペースを区切りとして単語を識別する。

注) `buffer` には 200 文字以下しか入っていないと仮定して良い。

例)

`buffer` : This is a pen

`note[0]` : This¥0

`note[1]` : is¥0

`note[2]` : a¥0

`note[3]` : pen¥0

このプログラムを作成した者のみ、次頁以降を読むこと。

このプログラムの中には、以下の3つの状態があることを考慮して、以下の議論を読んでほしい。最初はわかりにくい部分もあるかも知れないが、想像力を働かせて読んで欲しい。

状態（1）`note` を改行した、

状態（2）`note` に文字を格納した

状態（3）終了状態

議論 1 -----

`buffer` には `This is a pen` が

入っているとし、さらに、`k=3` とする。

`k=2` までの段階で、`note` には、以下のような文字が書かれている。

`note[0]` Thi

`note[1]`

`note[2]`

:

:

さて、`buffer[k]='s'` なので

`note[0][3]` には's'と記入する。

すなわち、状態が（2）で、入力が「文字」

であったときには次の動作を行う必要がある。

動作(a) `note` に `buffer` の文字を代入する

```
note[i][j]=buffer[k];
```

```
k++;
```

```
j++;
```

さらに、このとき、次の状態はもちろん、（2）となる。

議論 2 -----

buffer には This is a pen¥0

が入っているとし、さらに、k=4 とする。

k=3 までの段階で、note には、以下のような文字が書かれている。

note[0] This

note[1]

note[2]

:

:

さて、buffer[k]=' ' なので

note[0][3] には'¥0'と記入する。

(次の文字'i'は note[1][0]に書かねばならない)

すなわち、状態が (2) で、入力が「空白」

であったときには次の動作を行う必要がある。

動作(b) note に¥0 を代入し、次の単語を書くために改行する。

note[i][j]='¥0';

k++;

i++;

j=0;

さらに、次の状態はもちろん、(1) となる。

議論 3 -----

buffer には This is a pen¥0

が入っているとし、さらに、k=13 とする。

k=12 までの段階で、note には、以下のような文字が書かれている。

note[0] This¥0

note[1] is¥0

note[2] a¥0

note[3] pen

さて、`buffer[k]='¥0'` なので
`note[3][3]` には'¥0'と記入する。

すなわち、状態が（２）で、入力が「¥0」
であったときには次の動作を行う必要がある。

動作(c) `note` に¥0 を代入し、強制終了させる

```
note[i][j]='¥0';  
k=200;
```

さらに、次の状態はもちろん、（３）となる。

プログラムの状態変化について
上の議論をまとめると、以下のような表になる。

		今読み込んだもの（ <code>buffer[k]</code> ）		
		スペース	文字	'¥0'
い	（１）	？	？	？
ま				
の	（２）	（１）	（２）	（３）
状				
態	（３）	？	？	？

問：この表の残りの部分を埋めよ

また、プログラムの動作について
上の議論をまとめると、以下のような表になる。

		今読み込んだもの (buffer[k])		
		スペース	文字	'¥0'
い	(1)	?	?	?
ま				
の	(2)	(b)	(a)	(c)
状				
態	(3)	?	?	?

問：この表の残りの部分を埋めよ

次に、この表をプログラム化してみよう。

考察 1

まず、以下の関数をつくる。

関数名 value

引数 char c

戻り値

c=' ' ならば 0

c が文字ならば 1

c が'¥0'ならば 2

副作用 なし

考察 2

次のプログラムは上の 2 つの表をプログラム化しようとして作成したものであるが、不完全である。完全なものを作成せよ。

```
int main(void){
char buffer[200], note[10][20];
int i,j,k,now=1,next;
char action;

i=0;
```

```

j=0;
k=0;
while(k<=199){
    if (now==1&& value(buffer[k])==1) { action='a'; next=1;}
    if (now==1&& value(buffer[k])==2) { action='b'; next=2;}
    if (now==1&& value(buffer[k])==3) { action='c'; next=3;}
    .....

    if (action=='a') {note[i][j]=buffer[k]; k++; j++;}
    if (action=='b') {
        }
    .....

    now=next;
}
return 0;
}

```