

以下は、考察 2 の解答です。

```
int value(char c){
    if (c==' ') return 0;
    if (c=='¥0') return 2;
    return 1;
}

main(){
char buffer[200]="This is a pen", note[10][20];
int i,j,k,now=1,next;
char action;

for (i=0;i<10;i++) note[i][0]='¥0';
i=0;
j=0;
k=0;
    while(k<=199 ){
        if (now==1&& value(buffer[k])==0) { action='c'; next=1;}
        if (now==1&& value(buffer[k])==1) { action='a'; next=2;}
        if (now==1&& value(buffer[k])==2) { action='c'; next=3;}
        if (now==2&& value(buffer[k])==0) { action='b'; next=1;}
        if (now==2&& value(buffer[k])==1) { action='a'; next=2;}
        if (now==2&& value(buffer[k])==2) { action='c'; next=3;}
        if (now==3&& value(buffer[k])==0) { action='c'; next=3;}
        if (now==3&& value(buffer[k])==1) { action='c'; next=3;}
        if (now==3&& value(buffer[k])==2) { action='c'; next=3;}

        if (action=='a') {note[i][j]=buffer[k]; k++; j++;}
        if (action=='b') {note[i][j]='¥0';k++;i++;j=0;}
        if (action=='c') {note[i][j]='¥0';k=200;}

        now=next;
    }

printf("%s¥n",buffer);
for(i=0;note[i][0]!='¥0';i++) printf("%s¥n",note[i]);
}
```

注) このプログラムに於いて、while の条件を

now!=3

と変えても同じ動作をします。その場合、動作(c) は note[i][j]='¥0'のみです。

さて、上のプログラムで、main から数えて3行目で

now=1

としました (now=2 としても、同じ動作になります)。ここでは、1と2以外に代入する値が無かったため、やむを得ず1を代入しました。しかしながら、プログラムが、最初の文字を読んだときの状態は、本来、状態(1)でも状態(2)でもありません。そこで、初期状態(0)を新たに考えて、次のようなプログラムにします。

```
#include <stdio.h>
```

```
int value(char c){
```

```
    if (c==' ') return 0;
```

```
    if (c=='¥0') return 2;
```

```
    return 1;
```

```
}
```

```
main(){
```

```
    char buffer[200]="This is a pen", note[10][20];
```

```
    int i,j,k,now=0,next;
```

```
    char action;
```

```
    for (i=0;i<10;i++) note[i][0]='¥0';
```

```
    i=0;
```

```
    j=0;
```

```
    k=0;
```

```
        while(now!=3){
```

```
            if (now==0&& value(buffer[k])==0) { action='c'; next=1;}
```

```
            if (now==0&& value(buffer[k])==1) { action='a'; next=2;}
```

```
            if (now==0&& value(buffer[k])==2) { action='c'; next=3;}
```

```
            if (now==1&& value(buffer[k])==0) { action='c'; next=1;}
```

```
            if (now==1&& value(buffer[k])==1) { action='a'; next=2;}
```

```
            if (now==1&& value(buffer[k])==2) { action='c'; next=3;}
```

```
            if (now==2&& value(buffer[k])==0) { action='b'; next=1;}
```

```
            if (now==2&& value(buffer[k])==1) { action='a'; next=2;}
```

```

        if (now==2&& value(buffer[k])==2) { action='c'; next=3;}
        if (now==3&& value(buffer[k])==0) { action='c'; next=3;}
        if (now==3&& value(buffer[k])==1) { action='c'; next=3;}
        if (now==3&& value(buffer[k])==2) { action='c'; next=3;}

        if (action=='a') {note[i][j]=buffer[k]; k++; j++;}
        if (action=='b') {note[i][j]='\0';k++;i++;j=0;}
        if (action=='c') {note[i][j]='\0';}

        now=next;
    }
    printf("%s¥n",buffer);
    for(i=0;note[i][0]!='¥0';i++) printf("%s¥n",note[i]);
}

```

上のプログラムをさらに洗練させましょう。main 文中の

```

        if (now==0&& value(buffer[k])==0) { action='c'; next=1;}
        .....

        if (now==3&& value(buffer[k])==3) { action='c'; next=3;}

```

の部分を、2つの2次元配列を使えば、次のようにシンプルに書けます。

```

int value(char c){
    if (c==' ') return 0;
    if (c=='¥0') return 2;
    return 1;
}

main(){
int states[4][3]= {
{1,2,3},
{1,2,3},
{1,2,3},
{3,3,3}
};
char actions[4][3] = {
{'c','a','c'},
{'c','a','c'},
{'b','a','c'},
{'c','c','c'}
};
char buffer[200]="This is a pen", note[10][20];
int i,j,k,now=0,next;
char action;

for (i=0;i<10;i++) note[i][0]='¥0';
i=0;j=0;k=0;
    while(now!=3 ){
        action= actions[now][value(buffer[k])];
        next= states[now][value(buffer[k])];
        if (action=='a') {note[i][j]=buffer[k]; k++; j++;}
        if (action=='b') {note[i][j]='¥0';k++;i++;j=0;}
        if (action=='c') {note[i][j]='¥0';}
        now=next;
    }
printf("%s¥n",buffer);
for(i=0;note[i][0]!='¥0';i++) printf("%s¥n",note[i]);
}

```

練習問題 1：上のプログラムの動作を状態遷移図で表してみよう。

解答例はファイル `state_zu.pdf` の図 1。

練習問題 2：上のプログラムを改良して、「スペースが 2 個以上入力されたら読み飛ばす」プログラムを作れ。具体的には、①新しい `action (d)` を定義する。`(d)` の内容は `k++` のみ ②`actions` という 2 次元配列の適切な部分に `d` を書き入れる。

練習問題 3：

(1) 練習問題 1 のプログラムを改良して、キーボードから入力された英文を処理するプログラムを作れ (`gets` を使ってよい)。

(2) 以下の指示通り操作せよ。

(2-0) 上のプログラムの実行ファイルをつくる。(仮に `a.out` とする)。

(2-1) 適当な英文テキストファイル “`document`” を作成する。

(2-2) プログラム `con` (同じディレクトリにある `document` というファイルを 1 行表示する) を使って以下のコマンドを実行せよ。

```
./con | ./a.out
```

練習問題 4：2 桁 2 進数カウンタを作製するため、状態遷移図 (`state_zu.pdf` の図 2) を描いた。図中の記号 `1/a` は、「1 が入力されたときに動作 `a` を行う」という意味である。この図に対応する C 言語プログラムを作成せよ。ただし、図中の動作 `a,b,c,d` は以下のように定義する。

`a : printf("00¥n");`

`b : printf("01¥n");`

`c : printf("10¥n");`

`d : printf("11¥n");`

練習問題 5：C 言語のコメントを抜き出すプログラムを作るため、以下の問題を解け。

(1) 状態遷移図を作成せよ。なるべくノーヒントで解くことが望ましいが、分からない場合は、ファイル `hint_for_rensyuu5.pdf` を見よ。

(2) 状態遷移図に沿って C 言語プログラムを作成せよ。