＜演習問題＞　ポインタ(2)

【演習問題 1】

以下のプログラムは, 配列 moji1 に文字列”ABC”, 配列 moji2 に文字列”DEF”を格納して, 配列 moji1 と moji2 を連結した文字列を配列 moji1 に格納するプログラムとなる. プログラムを完成させて実行せよ.

```
#include<stdio.h>
void fstrcpy(char *p1, char *p2);   //文字列のコピー
void fstrcat(char *p1, char *p2);   //文字列の連結

int main(void)
{
    char moji1[10], moji2[10];

    fstrcpy(moji1, "ABC");   // moji1 に”ABC”を格納
    fstrcpy(moji2, "DEF");   // moji2 に”DEF”を格納
    fstrcat(moji1, moji2);     // moji1 の後ろに moji2 を連結

    printf(“moji1：%s¥n”, moji1);   // moji1 を表示
    printf(“moji2：%s¥n”, moji2);   // moji2 を表示

    return(0);
}

//  文字列のコピー, 文字列 p2 を p1 にコピー
void fstrcpy(char *p1, char *p2)
{
    // p1 に p2 を 1 文字ずつコピー
    while(*p2   !=   ’¥0’)
    {
        [                    ]
    }
    [            ]  // p1 の末尾にヌル文字を代入
}

//  文字列の連結, 文字列 p2 を文字列 p1 の後ろに連結
void fstrcat(char *p1, char *p2)
{
    // p1 の末尾まで(ヌル文字が表れるまで)ポインタの指す位置を動かす
    while( [          ] )
    {
        [            ]
    }

    // p1 の末尾に p2 を 1 文字ずつコピーして, p1 の末尾にヌル文字を代入
    [                    ]
}
```

実行結果
moji1：ABCDEF
moji2：DEF

【演習問題 2】

以下のプログラムは，配列name[] に格納された文字列“Nihon University”からスペース記号の位置を探し，スペース記号の左側の文字列をmalloc関数で確保した領域に保存して，表示するプログラムである．プログラムを完成させて実行せよ．

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

int main(void){
    int i;
    char name[] = "Nihon University";
    char *p;
    char *str;
    int num = 0; //  スペース記号の位置を格納

    //  スペース記号の位置を調べる
    p=name;
    while( *p != '¥0' ){
        num++;
        if( *p == '   ' ){
            printf("%d 文字目に見つかりました！¥n",      );
            break;
        }
        
    }

    //  スペース記号の左側を保存する領域を確保（文字列+'¥0'の領域）
    str = (char*)malloc(       * sizeof(char));

    //  スペース記号の左側の文字列を malloc で確保した領域にコピー
    for(i=0; i <       ; i++){
        
        = name[i];
    }
    
    = '¥0'; //終端記号

    //  スペース記号の左側の文字列を画面に表示
    printf("%s¥n", str);

    // malloc で確保した領域の解放
    

    return 0;
}
```

実行結果
6 文字目に見つかりました！
Nihon

【演習問題 3】

演習問題 2 のプログラムでは，char name[]="NihonUniversity"(スペース記号を削除)とした場合，画面には NihonUniversity と表示してしまう．そこで，配列 name にスペース記号が含まれている場合は，演習問題２と同様の動作をし，スペース記号が含まれていない場合は「スペース記号は見つかりませんでした」と表示して終了するようにプログラムを変更せよ．

なお，変更の際，以下の要件を満たす find 関数を作成し，main 関数から呼び出して，スペース記号の位置を調べること．

int find(char *s, char c);

引数

・s　：検索対象文字列

・c　：検索する文字

戻り値

・指定した文字が存在した場合　：　文字の位置を返す

・指定した文字が存在しない場合：　-1 を返す

【演習問題 4】

英単語（英文字で最大１５文字）を＊が現われるまで一つずつ読み込み，読込んだ英単語の文字数を計算して同一文字数の単語の出現回数を表示するプログラムを作成せよ。図の例を参考に，文字列は単語毎にキーボード入力するプログラムを作成すること。

（注意）　空白やピリオド、改行　は英単語の区切り記号である

ただし，プログラムは、

（１）英単語を格納する領域は文字型配列を使用して宣言して，実行文中でもその配列を使用するもの

（２）英単語を格納する領域は文字型配列を使用して宣言して，実行文中では配列を使用しないで配列の先頭番地を記憶するポインタ変数を用意して使用するもの

の２種類を作成せよ。

課題データ

The executable UML is a graphical specification language.
It combines a subset of the UML graphical notation with executable semantics and timing rules. ＊

英単語の入力例：

1 番目の英単語を入力してください(15 文字以内)：
--> The
2 番目の英単語を入力してください(15 文字以内)：
--> executable
3 番目の英単語を入力してください(15 文字以内)：
--> UML
4 番目の英単語を入力してください(15 文字以内)：
--> is
5 番目の英単語を入力してください(15 文字以内)：
--> a
6 番目の英単語を入力してください(15 文字以内)：
--> graphical
7 番目の英単語を入力してください(15 文字以内)：
-->

結果の表示法

文字数	出現回数
1	2
2	3
3	2
・	・
・	・
・	・
１５	1