

演習 1

次のプログラムの出力結果を、右に記入せよ

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

int main (void)
{
    char str1[100]="Intel",
        str2[100]="AMD",
        str3[100]="IBM";

    printf("%s\n%s\n%s\n\n", str1, str2, str3);

    strcat(str1, str3);
    strcpy(str3, str2);
    strncat(str3, str1, 3);

    printf("%s\n%s\n%s\n\n", str1, str2, str3);

    return 0;
}
```

演習 2

次のプログラムは、コンパイルエラーは出ないが、実行すると非常に危険である。
理由を述べよ。

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

int main (void)
{
    char str1[10]="Intel",
        str2[10]="Panasonic";

    printf("%s\n%s\n\n", str1, str2);

    strcat(str1, str2);

    printf("%s\n%s\n\n", str1, str2);

    return 0;
}
```

演習 3

strlen、strcpy、strcatと同じ働きをする関数mystrlen、mystrcpy、mystrcatを作れ。
なおプロトタイプ宣言は、都合により次の通りとする。

```
int mystrlen(char str1[])  
void mystrcpy(char str1[], char str2[])  
void mystrcat(char str1[], char str2[])
```

演習 4

次のプログラムは、ある文字列 (str1) に対し、入力によって得た文字列 (str2) を連結し、各アルファベット (大文字と小文字を区別しない) が、いくつ登場したか表示するプログラムである。空欄を補って完成させよ。

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

int main (void)
{
    char str1[20]="Intel",
          str2[10];
    int cnt[27]={0}, i=0;

    printf("文字列入力=>");
    scanf("%9s", str2);

    strcat(①_____);

    printf("%s¥n", str1);

    while(②_____ )
    {
        if(③_____ )
        {
            ④_____ ++;
        }
        else if('A' <= str1[i] && str1[i] <= 'Z' )
        {
            cnt[ ⑤_____ ]++;
        }
        else
        {
            printf("不正な入力文字です");
            return 0;
        }
        i++;
    }
    for(i=0; i<27; i++)
    {
        printf("%c:%d¥n", i+'A', cnt[i]);
    }

    return 0;
}
```

演習 5

演習 4 で作成したプログラムを、一個以上登場したアルファベットのみ表示するようにしたい。どこをどう弄ればよいか？ 下に記せ。



補足 2次元配列

```
int a[5][10];
```

上記のような宣言によって使える配列を、二次元配列と言う。これはさらに三次元、四次元と増やすことも出来る。意味は、

int型の要素10個を持つ配列をさらに5個、連続的に確保すると捕らえて欲しい。もし、一次元の配列で同様のことを実現しようとする、

```
int a[M][N];
```



```
int a[M*N];
```

と同等であり、またアクセス方法については、

```
a[i][j]
```



```
a[i*M+j]
```

が同等である。

また、この初期化は次のように書く。一行で書いても良いが、視覚的に分かりやすくするため、このような記述をすることが一般的である。

```
int a[2][3]={{1, 2, 3},
             {2, 6, 4}};

char str[3][100]={"Intel",
                 "AMD",
                 "IBM"};
```

演習 6

次のように初期化されて与えられる 2 次元配列がある。これを 3×3 の行列とみなし、この行列式の値を求めるプログラムを書け。

```
int a[3][3] = {{1, 2, 3},  
               {4, 5, 1},  
               {5, 2, 5}};
```


$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 1 \\ 5 & 2 & 5 \end{bmatrix}$$

演習 1 解答

```
Intel
AMD
IBM
```

```
IntelIBM
AMD
AMDInt
```

演習 2 解答

10個のスペースしかない配列str1に対し、明らかにそれ以上の書き込みをしようとしているため、本来確保されていない勝手な領域に書き込もうとしてしまうから。

演習 3 解答

str1、str2を実際に与え、元のstrlen、strcpy、strcatと動作を比較して確認する。

演習 4 解答

- ① str1, str2
- ② str1[i] != '\0'
- ③ 'a' <= str1[i] && str1[i] <= 'z'
- ④ cnt[str1[i] - 'a']
- ⑤ str1[i] - 'A'

演習 5 解答

```
printf("%c:%d\n", i+'A', cnt[i]);
```



```
if(cnt[i]!=0)
{
    printf("%c:%d\n", i+'A', cnt[i]);
}
```

演習 6 解答

手計算の結果 (= -58) と同じになればOK。

サラスを使ってやるなら、公式どおりのたった一行で答えが出る。

今後の発展のためには、余因子展開などの方法を利用したらどうなるか、また、 $n \times n$ 行の行列とした場合どうなるか検討してみると良い。