

## 自作問題 1

### 問題文

予め用意された配列の a と b の個数を数えるプログラム

### ソースコード

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

void string(const char *);

int main(void) {
    char a[3] = "Aaa";
    char *pa = a;
    char b[9] = "bbbbbbbbb";
    char *pb = b;
    char c[12];
    char *pc = c;
    strcpy(c, a);
    strcat(c, b);
    for (int i = 0; i < 11; i++) {
        printf("%c\n", c[i]);
    }
    string(pc);
}

void string(const char *c) {
    int num_a=0;
    int num_b=0;
    for(int i=0;i<11;i++){
        if (c[i]=='a'){
            num_a++;
        }
        else if (c[i]=='b'){
            num_b++;
        }
    }
}
```

```
}  
printf("a の数は%d\nb の数は%d",num_a,num_b);  
}
```

実行結果

```
A  
a  
a  
b  
b  
b  
b  
b  
b  
b  
b  
aの数は2  
bの数は8
```

## 問題文

$N$  個のオセロの石が一行に並んでいます。それぞれの石の状態は長さ  $N$  の文字列  $S$  によって表されており、 $S_i = B$  のとき左から  $i$  番目の石の表面は黒色、 $S_i = W$  のとき左から  $i$  番目の石の表面は白色となっています。

ここで、以下の操作を行うことを考えます。

- 左から  $i$  番目の石の表面が黒色、左から  $i+1$  番目の石の表面が白色であるような  $i$  ( $1 \leq i < N$ ) を一つ選び、その 2 つの石とともに裏返す。つまり、左から  $i$  番目の石の表面が白色、左から  $i+1$  番目の石の表面が黒色になるようにする。

最大で何回この操作を行うことができるか求めてください。

## 制約

- $1 \leq |S| \leq 2 \times 10^5$
- $S_i = B$  または  $W$

ソースコード

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

```
int main(void) {  
    int i;
```

```

char str[200000]={0};
int str_size=0;
scanf("%s",str);
for(i=0;i<200000;i++){
    if ((str[i]=="W")||(str[i]=='B')){
        str_size++;
    }
    else if (str[i]==0){
        break;
    }
}
char array[str_size];
for(i=0;i<str_size;i++){
    array[i]=str[i];
}
int result=0;
for(i=0;i<str_size;i++){
    if (array[i]=='B'){
        for(int j=i;j<str_size;j++){
            if (array[j]=="W"){
                result++;
            }
        }
    }
}
printf("%d",result);
return 0;
}

```

実行結果

**BBW**  
2

## 演習問題 1

### 問題文

次のような頭部をもつ、配列から配列へ  $t$  個の要素をコピーする関数を作れ。配列の長さは十分に用意されているとする。 `void ary_copy(char*d, char* s, int t)`

### ソースコード

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

void ary_copy(char *d, const char *s, int t);

int main(void) {
    int t = 5;
    char d[8] = {0};
    char s[8] = {"aiueo700"};
    ary_copy(d, s, t);

    return 0;
}

void ary_copy(char *d, const char *s, int t) {
    for (int i = 0; i < t; i++) {
        d[i] = s[i];
    }
    for (int i = 0; i < t; i++) {
        printf("d[%d]=%c\n", i, d[i]);
    }
}
```

### 実行結果

```
d[0]=a
d[1]=i
d[2]=u
d[3]=e
d[4]=o
```

## 演習問題 2

### 問題文

次のような頭部をもつ、2つの配列 に格納された文字列を結合して、他の配列へコピーする関数を作れ。配列の長さは十分に用意 されているとする。

### ソースコード

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

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

```
void str_concat(char *d, const char *s1, const char *s2);
```

```
int main(void) {
```

```
    char d[100];
```

```
    char s1[] = "おい引きこもり！";
```

```
    char s2[] = "聞いてんのか！？";
```

```
    str_concat(d, s1, s2);
```

```
    return 0;
```

```
}
```

```
void str_concat(char *d, const char *s1, const char *s2) {
```

```
    strcpy(d,s1);
```

```
    strcat(d,s2);
```

```
    printf("%s",d);
```

```
}
```

### 実行結果

おい引きこもり！聞いてんのか！？

### 演習問題 3

#### 問題文

関数外部で宣言された要素数  $n$  の `int` 型配列の先頭要素のアドレスを受け取り、その配列に格納されている整数値で最大のものを格納している 配列要素のアドレスを返却する関数 `max_element` を作れ。以下の関数頭部を持つとする。配列に格納されている整数値はすべて異なると 仮定してよい。

#### ソースコード

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

```
int *max_element(int *ptr, int n);
```

```
int main(void) {  
    const int n = 10;  
    int *return_ptr;  
    int str[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};  
    int *ptr;  
    ptr = str;  
    return_ptr = max_element(ptr, n);  
    printf("最大値のアドレスは%p\n", return_ptr);  
    printf("最大値は%d", *return_ptr);  
    return 0;  
}
```

```
int *max_element(int *ptr, int n) {  
    int max;  
    int *max_ptr;  
    max = *ptr;  
    max_ptr = ptr;  
    ptr++;  
    for (int i = 0; i < n; i++) {  
        if (max < *ptr) {  
            max = *ptr;  
            max_ptr = ptr;  
        }  
        ptr++;  
    }
```

```
}  
    return max_ptr;  
}
```

## 実行結果

最大値のアドレスは0x7ffee4dadab4

最大値は10



## 演習問題4

### 問題文

関数外部で宣言された要素数  $n$  の `double` 型配列を3つ考える。それら3つの配列らの先頭要素のアドレスを受け取り、1番目と2番目の配列の  $i$  番目の要素に格納されている値の和を3番目の配列の  $i$  番目の要素に格納する関数 `vector_add` を作れ。以下の関数頭部を持つとする。

### ソースコード

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

```
void vector_add(const double *ptr1, const double *ptr2, double *ptr3);
```

```
int main(void) {  
    double str1[5] = {10, 20, 30, 40, 50};  
    double str2[5] = {10, 20, 30, 40, 50};  
    double str3[5] = {0};  
    vector_add(str1, str2, str3);  
    return 0;  
}
```

```
void vector_add(const double *ptr1, const double *ptr2, double *ptr3) {  
    for(int i=0; i<5; i++){  
        *(ptr3+i) = *(ptr1+i) + *(ptr2+i);  
        printf("% .1f", *(ptr3+i));  
    }  
}
```

### 実行結果

```
20.0 40.0 60.0 80.0 100.0
```