

プログラミング基礎 前期期末試験

全てのプログラムファイルの先頭行に、コメントとして自分の番号と名前を書くこと。

- 1 int 型の変数 x を宣言し、scanf() で入力された値を変数 x に代入し、「 2^x 」を計算して出力するプログラムを main() 内に作りなさい。ただし、この計算には**シフト演算**を使うこと。実行結果は以下のようになる。

```
[実行結果]
$ ./a.out      (← x に 8 を代入した場合の実行)
x > 8
2 の 8 乗は 256
$ ./a.out      (← x に 1 を代入した場合の実行)
x > 1
2 の 1 乗は 2
$ ./a.out      (← x に 0 を代入した場合の実行)
x > 0
2 の 0 乗は 1
```

- 2 char 型の配列 str1 を以下のようにグローバル変数として宣言する。

```
char str1[50]="Today is 2017/8/1";
```

この str1 に格納された文字列に対して、「文字列に含まれる数字, 英文字, それ以外の文字数を数えて出力する」関数 show_count() を作成しなさい。この関数には、**標準ライブラリ関数**を利用してよい。この関数のプロトタイプ宣言は以下のようになる。

```
void show_count();
/* 数字, 英文字, それ以外の文字の数を数える変数をそれぞれ用意する */
/* str1 に格納されている文字列に対して繰り返し処理する */
/* 繰り返し処理の中で、数字, 英文字, それ以外を判別し、それぞれの個数をカウントしていく */
/* 最後に、それぞれの個数を出力する（出力の様子は実行結果を参照） */
/* 数字かどうかの判別には、関数 isdigit() を使う */
/* 英文字かどうかの判別には、関数 isalpha() を使う */
```

main() での動作確認の例とその実行結果は以下のようになる。

```
[main での処理]
show_count();
[実行結果]
数字: 6 個, 英文字: 7 個, その他: 4 個
```

- 3 int 型の多次元配列を以下のようにグローバル変数として宣言する。

```
#define X 3    /* 行 */
#define Y 4    /* 列 */
int code[X][Y] = {
    {65, 66, 88, 89},
    {50, 51, 53, 55},
    {43, 45, 60, 62}
};
```

この多次元配列 code の指定した行に対して、「各要素の値を、10 進数, 文字 (ASCII コード表上の文字), 16 進数で出力する」関数 show_code() を作成しなさい。この関数のプロトタイプ宣言は以下のようになる。

```
void show_code(int x);  
/* 引数 x で指定された行に対する繰り返し処理を作る */  
/* 繰り返し処理の中で、この行の各要素の値を、10 進数、文字、16 進数で出力する */
```

main() での動作確認の例とその実行結果は以下のようになる。

```
[main での処理]  
show_code(0);    /* 0 行目に対する処理 */  
show_code(1);    /* 1 行目に対する処理 */  
show_code(2);    /* 2 行目に対する処理 */  
[実行結果]  
65 A 41          (← 0 行目に対する処理)  
66 B 42  
88 X 58  
89 Y 59  
50 2 32          (← 1 行目に対する処理)  
51 3 33  
53 5 35  
55 7 37  
43 + 2b          (← 2 行目に対する処理)  
45 - 2d  
60 < 3c  
62 > 3e
```

4 次のように年月日を 8 桁で表す整数を考える。

「年を西暦で表す 4 桁」と「月を表す 2 桁」と「日を表す 2 桁」をつなげて→ 合計 8 桁の整数
(例) 2017 年 8 月 1 日 → 「20170801」という 8 桁の整数で表す

このような年月日を表す整数を引数で受け取り、この整数を「年」「月」「日」に分けて出力する関数 show_date() を作成しなさい。この関数のプロトタイプ宣言は以下のようになる。

```
void show_date(int date);  
/* まず、date と 10000 の商（これが「年」になる）を出力する */  
/* date と 10000 の余りを求め、下 4 桁のみにする */  
/* date と 100 の商（これが「月」になる）を出力する */  
/* …という処理を「日」の出力まで続ける */
```

main() での動作確認の例とその実行結果は以下のようになる。

```
[main での処理]  
show_date(20170811);    /* 2017 年 8 月 11 日を表す 8 桁の整数を引数で渡す */  
[実行結果]  
2017 年 8 月 11 日
```

5 char 型の配列 str2 を以下のようにグローバル変数として宣言する。

```
char str2[50]="abyz#$$%0189";
```

この str2 に格納された文字列に対して、「英文字（アルファベット大文字と小文字両方とも）と数字をハイフン (-) に変換する」関数 replace_alnum を作成しなさい。ただし、**この関数には、標準ライブラリ関数を使わないこと**。この関数のプロトタイプ宣言は以下のようになる。

```
void replace_alnum();  
/* str2 に格納されている文字列に対して繰り返し処理する */  
/* 繰り返し処理の中で、i 番目の文字が、小文字のアルファベットか、大文字のアルファベットか、数字であるかを調べる */  
/* (数字は'0'~'9' の範囲にあるかどうかで判別できる) */  
/* 上記の条件を満たした場合は、そこにハイフンを代入する */
```

main() での動作確認の例とその実行結果は以下のようになる。

```
[main での処理]  
replace_alnum();  
printf("str2: %s\n", str2);  
[実行結果]  
str2: ----#%-----
```

問題はここまで

1 ~ 5 各 20 点

定期試験の実施について

試験中に使用できるもの

- 筆記用具（メモ用紙は必要な人に配布）
- 演習室のコンピュータ一台（一つの机に一人の配置で、座る場所はどこでもよい）

試験中に参照できるもの

- 自分のホームディレクトリ（ホームフォルダ）以下に保存されているファイル
（定期試験では紙媒体のものは参照不可）
- * 上記以外の情報を参照することは不正行為とする
（例：USB で接続された機器に保存されているファイルの参照など）
- * 試験中は、演習室外へのネットワークアクセスは遮断される

答案の提出

- 提出する全てのプログラムファイルの先頭行に、自分の学科の出席番号と氏名をコメントとして書く
- 保存したファイルは次のように「report」コマンドで提出する
（ちゃんと提出できた場合は、「Succeed.」と画面に表示される）
\$ ~kogai/report kiso1term 「プログラムファイル」
- 複数のファイルを提出する場合は、report コマンドを分けて提出する
例えば、test1.c と test2.c のファイルを提出したい場合は、次のように 2 回に分けて提出する

```
$ ~kogai/report kiso1term test1.c  
$ ~kogai/report kiso1term test2.c
```

- 同じ問題に対して、複数の提出ファイルが存在した場合は、更新日時が新しい方を提出ファイルとする
- 提出するファイルは、誰から提出されたのか区別されるため、ファイル名は各自で自由に決めて良い

前期中間試験 模範解答 (平均 87.5 点)

採点について コンパイル時にエラーとなる箇所は -4 点, 実行可能だが処理内容が問題の意図と違う箇所は -2 点を基本とする。

配点: 1 ~ 5 各 20 点

```
/* 自分の番号と名前を書く */
#include <stdio.h>
#include <ctype.h>

/* 関数のプロトタイプ宣言 */
void show_count();
void show_code(int x);
void show_date(int date);
void replace_alnum();

/* 多次元配列の宣言 */
#define X 3 /* 行 */
#define Y 4 /* 列 */
int code[X][Y] = {
    {65, 66, 88, 89},
    {50, 51, 53, 55},
    {43, 45, 60, 62}
};

/* 文字列の宣言 */
char str1[50]="Today is 2017/8/1";
char str2[50]="abyz##%0189";

/* [2] 文字列に含まれた数字・英文字をカウントする */
void show_count()
{
    int i;
    int count1 = 0, count2 = 0, count3 = 0;
    for(i=0; str1[i]!='\0'; i++) {
        if(isdigit(str1[i])) {
            count1++;
        }
        else if(isalpha(str1[i])) {
            count2++;
        }
        else {
            count3++;
        }
    }
    printf("数字: %d 個, 英文字: %d 個, その他: %d 個\n",
           count1, count2, count3);
}

/* [3] 多次元配列に格納された文字コードを出力する */
void show_code(int x)
{
    int i;
    for(i=0; i<Y; i++) {
        printf("%d %c %x\n", code[x][i], code[x][i],
               code[x][i]);
    }
}
```

```
/* [4] 日付を表す整数から年月日を出力する */
void show_date(int date)
{
    /* int year, month, day; */
    /* day = date % 100; */
    /* date /= 100; */
    /* month = date % 100; */
    /* date /= 100; */
    /* year = date; */
    /* printf("%d年%d月%d日\n", year, month, day); */
    printf("%d年", date/10000);
    date %= 10000;
    printf("%d月", date/100);
    date %= 100;
    printf("%d日\n", date);
}

/* [5] 文字列の英数字を置き換える */
void replace_alnum()
{
    int i;
    for(i=0; str2[i]!='\0'; i++) {
        if( (str2[i] >= 'a' && str2[i] <= 'z') ||
            (str2[i] >= 'A' && str2[i] <= 'Z') ||
            (str2[i] >= '0' && str2[i] <= '9')) {
            str2[i] = '-';
        }
    }
}

int main(void)
{
    int x;
    /* [1] 2 の x 乗をシフト演算で求める */
    printf("x > ");
    scanf("%d", &x);
    printf("2 の %d 乗は %d\n", x, 1 << x);
    /* show_count() の動作確認 */
    show_count();
    /* show_code() の動作確認 */
    show_code(0);
    show_code(1);
    show_code(2);
    /* show_date() の動作確認 */
    show_date(20170811);
    /* replace_alnum() の動作確認 */
    replace_alnum();
    printf("str2: %s\n", str2);

    return 0;
}
```