

プログラミング基礎 前期中間試験

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 main() にて、scanf() で入力した整数 a に対して「1 から a までの整数の平方根を出力する」プログラムを作成しなさい。実行結果は以下のようになる。

```
[実行結果 その 1]
a > 3      (← 3 を入力した)
1.000000
1.414214
1.732051
[実行結果 その 2]
a > 9      (← 9 を入力した)
1.000000
1.414214
1.732051
2.000000
2.236068
2.449490
2.645751
2.828427
3.000000
```

- 2 int 型の配列 num1 を次のようにグローバル変数として宣言する。

```
#define N 5
int num1[N] = {3, 6, 5, 1, 2};
```

この配列 num1 に対して、「偶数となる要素の値を出力する」関数 show_even() 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_even();
/* 配列 num1 に対する繰り返し処理をする */
/* 繰り返し処理の中で、要素の値が偶数かを調べ、偶数の場合は出力する */
/* 出力の様式は実行結果を参照 */
/* 偶数かどうかは、「2 で割った余りが 0 かどうか」で判別できる */
```

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_even();
num1[0] = 10;
num1[3] = 20;
show_even();
[実行結果]
num1[1]: 6      (←最初の num1 に対する出力)
num1[4]: 2
num1[0]: 10     (← 0 番目と 3 番目に偶数を代入した後の num1 に対する出力)
num1[1]: 6
num1[3]: 20
num1[4]: 2
```

- 3 (1) 「引数 x の値が正 (0 より大きい) ならば 1、そうでないなら 0 とする」マクロ ISPOS を定義し、main() で動作を確認するプログラムを作成しなさい。

[main() での処理]

```
printf("%d\n", ISPOS(5));  
printf("%d\n", ISPOS(-2));  
printf("%d\n", ISPOS(0));
```

[実行結果]

```
1          (← 5 は正だから 1 となる)  
0          (← -2 は負だから 0 となる)  
0          (← 0 は負の扱いとなり 0 となる)
```

- (2) int 型の配列 num2 を次のようにグローバル変数として宣言する。

```
#define N 5  
int num2[N] = {4, -1, -9, 3, -2};
```

この配列 num2 に対して、「正の値となる要素数を返す」関数 count_pos() を作成しなさい。
ただし、先に作ったマクロ ISPOS を使うこと。この関数のプロトタイプ宣言は以下ようになる。

```
int count_pos();  
/* 配列 num2 に対する繰り返し処理を作る */  
/* 繰り返し処理の中で、マクロ ISPOS の結果を加算して、正の値の個数をカウントする */  
/* 求めた個数を戻り値とする */
```

main() での動作確認の例とその実行結果は以下ようになる。

[main での処理]

```
printf("%d\n", count_pos());  
num2[1] = 10;  
printf("%d\n", count_pos());  
num2[4] = 20;  
printf("%d\n", count_pos());
```

[実行結果]

```
2          (← 正の値の要素数は 2 個)  
3          (← 1 番目の要素を 10 に変更した後は 3 個になる)  
4          (← 4 番目の要素を 20 に変更した後は 4 個になる)
```

- 4 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] = "Programming Language";
```

この配列 str1 に対して、「引数に指定した文字が含まれている文字数を返す」関数 count_char() を作成しなさい。
この関数のプロトタイプ宣言は以下ようになる。

```
int count_char(char ch);  
/* 配列 str1 に格納されている文字列に対する繰り返し処理を作る */  
/* 繰り返し処理の中で、引数 ch と各文字を比較し、等しい場合は個数をカウントする */  
/* 求めた文字数を戻り値とする */
```

main() での動作確認の例とその実行結果は以下ようになる。

[main での処理]

```
char c;  
printf("c > ");  
scanf("%c", &c);  
printf("count_char: %d\n", count_char(c));
```

[実行結果 その 1]

```
c > a          (←文字 a を入力)  
count_char: 3  (← a は 3 文字含まれている)
```

[実行結果 その 2]

```
c > r          (←文字 r を入力)  
count_char: 2  (← r は 2 文字含まれている)
```

問題はここまで

1 ~ **4** 各 25 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ー一台（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
- * 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 kiso1mid 「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 kiso1mid test1.c
```

```
$ ~kogai/report kiso1mid 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 86.2 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 ~ 4 各 25 点

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

/* 配列をグローバル変数として宣言 */
#define N 5
int num1[N] = {3, 6, 5, 1, 2};
int num2[N] = {4, -1, -9, 3, -2};
char str1[50] = "Programming Language";

/* [3] 正数かどうか判断するマクロ */
#define ISPOS(x) (x > 0 ? 1 : 0)

/* 関数のプロトタイプ宣言 */
void show_even();
int count_pos();
int count_char(char ch);

/* [2] 偶数のみの要素を出力する */
void show_even()
{
    int i;
    /* 配列 num1 に対する繰り返し処理 */
    for(i=0; i<N; i++) {
        /* 偶数かどうかを調べる */
        if(num1[i] % 2 == 0) {
            /* 偶数の場合は値を出力する */
            printf("num1[%d]: %d\n", i, num1[i]);
        }
    }
}

/* [3] 正数の要素数を返す */
int count_pos()
{
    int i, r;
    r = 0;
    /* 配列 num2 に対する繰り返し処理 */
    for(i=0; i<N; i++) {
        /* マクロ ISPOS の結果を加算する */
        r = r + ISPOS(num2[i]);
    }
    /* 全ての要素に対して加算した結果を返す */
    return r;
}

/* [4] 指定した文字が含まれている個数を返す */
int count_char(char ch)
{
    int i, r;
    r = 0;
    /* 文字列 str1 に対する繰り返し処理 */
    for(i=0; str1[i]!='\0'; i++) {
        /* ch と等しいか調べて個数を数える */
        if(str1[i] == ch) r++;
    }
    /* 数えた個数を返す */
    return r;
}

int main(void)
{
    int a, i;
    char c;

    /* [1] 1~a の平方根を出力する */
    /* a を入力する */
    printf("a > "); scanf("%d", &a);
    /* 1~a までの繰り返し処理をする */
    for(i=1; i<=a; i++) {
        /* i の平方根を出力する */
        printf("%lf\n", sqrt(i));
    }
    /* [2] show_even() の動作確認 */
    show_even();
    num1[0] = 10;
    num1[3] = 20;
    show_even();
    /* [3] ISPOS の動作確認 */
    printf("%d\n", ISPOS(5));
    printf("%d\n", ISPOS(-2));
    printf("%d\n", ISPOS(0));
    /* [3] count_pos() の動作確認 */
    printf("%d\n", count_pos());
    num2[1] = 10;
    printf("%d\n", count_pos());
    num2[4] = 20;
    printf("%d\n", count_pos());
    /* [4] count_char() の動作確認 */
    printf("c > ");
    scanf("%c", &c);
    printf("count_char: %d\n", count_char(c));

    return 0;
}
```