

**プログラミング
中間テスト（ペーパーテスト）問題用紙
（2021年12月03日）**

解答上、その他の注意事項

1. 問題は、問 I ~ V までである。
2. 持ち込みは不可である。筆記用具・時計・学生証以外のものは、かばんの中などにしまうこと。
3. 解答用紙の右上の欄に学籍番号・名前を記入すること。
4. 解答欄がマス目になっている場合は、1字に1マスを用いること。特に空白にも必ず1マスを用いること。
5. 解答中の文字 (特に a と d) がはっきりと区別できるよう注意すること。
6. スペースの都合上、テスト中のプログラム例では、最初の `#include <stdio.h>` は省略する。

- I. 以下のプログラムは動作はするがインデントーションが減茶苦茶でプログラムの構造を正しく反映していない。(そして、おそらく作成者の意図したようには動かない。)プログラムの構造を明確にするために「プログラミング」のインデントーションの約束事(別紙)に厳密に従って(特に5番目のルールに注意)、必要に応じてブレースを挿入し、インデントしなせよ。ただし、ブレース内の字下げは紙幅の制限のため3文字分とせよ。

もちろん、プログラムの意味(実行結果)が変わらないようにすること。(つまり、作成者の(よくわからない)意図を忖度するのではなく、コンパイラーが解釈する通りにインデントすること。)

解答は、最初の#include <stdio.h>は省略せよ。また、解答欄のスペースの都合で、(最後は別として)空行は入れずに記入せよ。

```
1 int main(void) {
2     int i = 0, j = 9;
3     while (j-- > 0)
4     while (i++ < 7)
5     if (i % 2 == 0)
6     putchar('*');
7     if (j % 3 == 0)
8     putchar('@');
9     else if (j % 5 == 0)
10    putchar('&');
11    putchar('#');
12    putchar('\n');
13    return 0; }
```

- II. 次のプログラムは一つの整数値を読み込んで、それを整数として3で割った値と、3.3倍した値を表示するプログラムである。たとえば、8を入力すると、

```
整数を入力して下さい: 8
それを整数として 3 で割った値は 2 です。
それを 3.3 倍した値は 26.400000 です。
```

と出力する。空欄を埋めて、プログラムを完成せよ。

```
1 int main(void) {
2     (i) m;
3     printf("整数を入力して下さい: ");
4     scanf("(ii)", (iii));
5     printf("それを整数として 3 で割った値は (iv) です.\n", m / 3);
6     printf("それを 3.3 倍した値は (v) です.\n", (vi));
7     return 0;
8 }
```

III. 次のプログラムの出力を書け。

```
1 int main(void) {
2     int n = 60;
3
4     if (n % 3 == 0) {
5         printf("A ");
6     }
7     if (n % 7 == 0) {
8         printf("B ");
9     } else if (n % 12 == 0) {
10        printf("C ");
11    } else if (n % 15 == 0) {
12        printf("D ");
13    }
14    putchar('\n');
15    return 0;
16 }
```

IV. 以下の空欄を埋めて、0 以上 300 未満の 3 の倍数をすべて昇順（つまり、0, 3, 6, ..., 297 の順）に出力するプログラムを完成させよ。

```
1 int main(void) {
2     int n;
3
4     for ( _____ ) {
5         printf("%d ", n);
6     }
7
8     putchar('\n');
9     return 0;
10 }
```

V. 以下のプログラムは、整数の組のデータを (i, j) という形式で出力する。このプログラムの出力を書け。ただし、この問題の解答では、空白は明示的に空白記号 (`␣`) を、改行は矢印 (`\n`) を書くこと。

```
1 int main(void) {
2     int i, j, n = 4;
3     for (i = n; i >= 1; i--) {
4         for (j = n; j >= i; j -= 2) {
5             printf("(%d,%d) ", i, j);
6         }
7         putchar('\n');
8     }
9     return 0;
10 }
```

「プログラミング」インデントーションについての約束事

<http://guppy.eng.kagawa-u.ac.jp/2021/Programming/indentation.html>より抜粋したものである。(ただし今回の問題に関係ない部分は割愛した。また、字下げ幅はオリジナルは4だが、ここでは紙幅の制限のため、3とした。)

0. 原則として、一行には文は一つしか書かない。
1. ブレース (`{ ~ }`) の中の文は、外よりも 3 字分を字下げする。
2. 一つのファイルで空白文字とタブ文字を混ぜない。(タブ文字を使わずに空白文字だけで字下げする。あるいは空白文字を使わずにタブ文字だけで字下げする。)
3. 開きブレース (`{`) は `if` や `else`, `switch`, `do`, `while`, `for` などのキーワードと 同行に改行せずに書く。開きブレースのあとは何も書かず改行する。
4. 閉じブレース (`}`) は対応する `if` や `switch`, `do`, `while`, `for` などのキーワードのはじめの文字と列をそろえて書く。その行には閉じブレース以外には何も書かない。
 - ただし、`else ...` や `do ~ while` 文の `while ...` は閉じブレース (`}`) と同行に続けて書く。(`else` や `do ~ while` 文の `while` は、新しい文の始まりではないことに注意する。)
5. `if` 文, `if ~ else` 文 や `for` 文, `while` 文, `do ~ while` 文などでは、選択されたり、繰り返したりされる文が一つだけの場合も、ブレース (`{ ~ }`) に囲む。この約束事だけは、教科書のプログラム例は必ずしもそうになっていないので、特に注意する。
 - ただし、`else` の対象が一つの `if` 文の場合の `else if ...` というかたちは、`else { if ... }` とはしない。

良い

```
if (n > 0) {
    printf("正です。 \n");
} else if (n == 0) {
    printf("0です。 \n");
} else {
    printf("負です。 \n");
}
```

悪い

```
if (n > 0) {
    printf("正です。 \n");
} else {
    if (n == 0) {
        printf("0です。 \n");
    } else {
        printf("負です。 \n");
    }
}
```


