

プログラミング
中間テスト（ペーパーテスト）問題用紙
(2024 年 11 月 27 日)

解答上、その他の注意事項

1. 問題は、問 I ～ V まである。
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、I 字に I マスを用いること。特に空白にも必ず I 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が滅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 if 文、if ~ else 文、for 文は「インデントの約束事」（別紙）の 5.「鉄の掟」に従って、必ずブレースを使用するように（ただしプログラムの意味（実行結果）が変わらないように）書き換え、さらに「インデントの約束事」の 0. ~ 4. に従ってインデントしなおせ。

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;
3     for (i = 1; i <= 6; i++)
4         if (i % 2 == 1)
5             putchar('#');
6         else
7             putchar('$');
8         if (i % 3 == 2)
9             putchar('%');
10            putchar('&');
11    return 0;
12 }
```

- II. 次のプログラムは一つの値を読み込んで、それを 1.111 倍した値と、整数として 3 で割った値を表示するプログラムである。たとえば、100 を入力すると、

```
整数を入力して下さい: 100
入力値を 1.111 倍すると 111.100000
入力値を整数として 3 で割ると 33
```

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     int n;
3     printf("整数を入力して下さい: ");
4     scanf("(i)", (ii));
5     printf("入力値を 1.111 倍すると (iii) \n", (iv));
6     printf("入力値を整数として 3 で割ると (v) \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 = 9;
3
4     switch (n % 3) {
5         case 0: printf("S ");
6         case 1: printf("T ");
7         default: printf("U ");
8     }
9
10    switch (n % 4) {
11        case 0: printf("X "); break;
12        case 1: printf("Y "); break;
13        default: printf("Z ");
14    }
15    return 0;
16 }
```

IV. 以下の空欄を埋めて、100 以上 300 以下の 3 の倍数をすべて降順（つまり、300, 297, 294, …, の順）に出力するプログラムを完成させよ。

```
1 int main(void) {
2     int k;
3
4     for (_____ ) {
5         printf("%d ", k);
6     }
7     putchar('\n');
8
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 (↵) を書くこと。

```
1 int main(void) {
2     int i, j, n = 6;
3     for (i = n; i >= 0; i -= 2) {
4         for (j = i; j <= 2 * i; j += 3) {
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

<https://guppy.eng.kagawa-u.ac.jp/2024/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。）

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 4 字分を字下げする。
2. （問題に関係ないので、省略する）
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じ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最初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（}）と同じ行に続けて書く。（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. （鉄の掟）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");
    }
}
```

プログラミング・中間テスト解答用紙 (2024 年 11 月 27 日)

学籍番号		氏名	
------	--	----	--

I.

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

```
int i;
```

```
return 0;
```

}

II.

(i)

(ii)

(iii)

(iv)

(v)

(vi)

--

[illegible]

This image shows a full page of white paper designed for handwriting practice. It features 20 evenly spaced, horizontal dashed gray lines that run across the entire width of the page. There are no margins, text, or other markings present.