

受験番号 _____ 氏名 _____

問題 1

以下は C 言語において, for 文の中で printf 関数を使うことで数列を出力しようとしたものである (解答に関係しない include 文や以下の文を囲む関数の定義文などは省略してある). 例題を参考に, 各設問における下線を引いた空白部を埋めて指定された数列が出力されるようにせよ. ただし, 使ってよい変数は

```
int i, j;
```

の二つのみとする. また解答で使える文字は空白文字と英字 i, j, 任意の整数, 以下の記号のみとする.

+, -, *, /, %, (,), =

解答は次ページの解答欄に記入すること.

(例題) 出力: 1, 2, 3, 4, 5, 6, 7,

```
int i = 0, j = 0;
for (i = 1; i < 8; i++) {
    _____;
    printf("%d, ", j);
}
```

(設問 2) 出力: 1, 3, 6, 10, 15, 21, 28,

```
int i = 0, j = 0;
for (i = 1; i < 8; i++) {
    _____;
    printf("%d, ", j);
}
```

(設問 4) 出力: 1, -2, 3, -4, 5, -6, 7,

```
int i = 0, j = 0;
for (i = 1; i < 8; i++) {
    j = -1 + (i % 2) * 2;
    _____;
    printf("%d, ", j);
}
```

(設問 6) 出力: 2, 3, -4, -5, 6, 7, -8, -9,

```
int i = 0, j = 0;
for (i = 2; i < 10; i++) {
    _____;
    printf("%d, ", j);
}
```

(設問 1) 出力: 2, 4, 6, 8, 10, 12, 14,

```
int i = 0, j = 0;
for (i = 1; i < 8; i++) {
    _____;
    printf("%d, ", j);
}
```

(設問 3) 出力: 1, 0, 3, 0, 5, 0, 7,

```
int i = 0, j = 0;
for (i = 1; i < 8; i++) {
    _____;
    printf("%d, ", j);
}
```

(設問 5) 出力: 0, 0, 2, 2, 4, 4, 6, 6,

```
int i = 0, j = 0;
for (i = 0; i < 8; i++) {
    _____;
    printf("%d, ", j);
}
```

(設問 7) 出力: 1, 20, 3, 40, 5, 60, 7, 80, 9,

```
int i = 0, j = 0;
for (i = 1; i < 10; i++) {
    _____;
    printf("%d, ", j);
}
```

受験番号 _____ 氏名 _____

解答欄:

例題	$j = i$
設問 1	
設問 2	
設問 3	
設問 4	
設問 5	
設問 6	
設問 7	

受験番号 _____ 氏名 _____

問題 2

以下は C 言語のプログラムにおいて、変数定義と関数のプロトタイプ宣言、それらを使った呼び出し文を抜き出したものである。各設問における下線を引いた空白部を適切に埋め、型の不一致によるコンパイルエラーを起さないようにせよ。ただし、解答に使ってよいのは変数名と C 言語での型名、以下の記号のみとする。

(,), *, &

解答は次ページの解答欄に記入すること。

<p>(例題)</p> <pre>void f(_____); int v; f(v);</pre>	<p>(設問 1)</p> <pre>int f1(int); int v1; f1(_____);</pre>
<p>(設問 2)</p> <pre>int f2(int *); int *v2; f2(_____);</pre>	<p>(設問 3)</p> <pre>int f3(int *); _____ v3; f3(&v3);</pre>
<p>(設問 4)</p> <pre>int f4(_____); int **v4; f4(*v4);</pre>	<p>(設問 5)</p> <pre>int f5(int); double v5; f5(_____);</pre>
<p>(設問 6)</p> <pre>int f6(int *); double *v6; f6(_____);</pre>	<p>(設問 7)</p> <pre>int f7(_____); int v7[10]; f7(v7);</pre>
<p>(設問 8)</p> <pre>int f8(int); int v8[10]; f8(_____);</pre>	<p>(設問 9)</p> <pre>int f9(_____); int **v9[10]; f9(**(v9+3));</pre>
<p>(設問 10)</p> <pre>int f10(int); struct _S10 { int *v; /* この値を使うこと */ }; struct _S10 *s10; f10(_____);</pre>	<p>(設問 11)</p> <pre>int f11(int (*)(int, int)); _____ ; f11(v11);</pre>

受験番号 _____

氏名 _____

解答欄：

例題	int
設問 1	
設問 2	
設問 3	
設問 4	
設問 5	
設問 6	
設問 7	
設問 8	
設問 9	
設問 10	
設問 11	

受験番号 _____ 氏名 _____

問題 3

N 人の名前が、下図の例のように、すべて小文字のアルファベットで N 行 M 列の 2 次元配列 `name[][]` に格納されている。それぞれの名前の直後には、文字列の終端を示す NULL コード (`\0`) が置かれている。これらの名前を辞書順に画面に表示したい。辞書順とは、たとえば “aa” は “ab” よりも上位であり、“a” は “aa” よりも上位であるような順序である。

以下の設問においては、必要な変数は適宜宣言して使用して良い。ただし、C 言語の標準ライブラリとして用意されている文字列関数 (`strcpy()` や `strcmp()` など) は使用しないものとする。

	name[i][j]										
i=0	k	i	m	u	r	a	\0				
1	a	o	k	i	\0						
2	y	a	m	a	d	a	\0				
3											

設問 1

任意の文字列 s と t を与えたとき、辞書順でどちらが上位になるかを判定する関数 `strorder()` を、以下の空欄を埋めて作成せよ。ただし、関数 `strorder()` は、文字列 s が上位であるとき 1 を返し、文字列 t が上位であるとき -1 を返し、両者が同一であるとき 0 を返すものとする。

```
#include <stdio.h>

int strorder( char s[], char t[] );

int main(void)
{
    char s[10] = "kimura";
    char t[10] = "aoki";
    int ret;

    ret = strorder(s, t);

    if( ret > 0 )
        printf("s が上位\n");
    else if ( ret == 0 )
        printf("s と t は等しい\n");
    else
        printf("t が上位\n");

    return 0;
}

int strorder( char s[], char t[] )
{
    

ア


}

```

解答欄：

ア

受験番号 _____ 氏名 _____

(問題 3) 設問 2

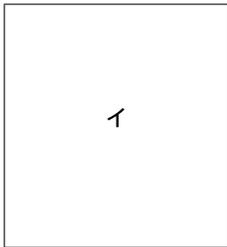
設問 1 で作成した関数 `strorder()` が利用できるものとし、以下の空欄を埋めて、2 次元配列 `name[][]` に格納された N 人の名前を辞書順に表示するプログラムを作成せよ。

```
#include <stdio.h>
#define N 5
#define M 10

int strorder( char s[], char t[] );

int main(void)
{
    char name[N][M] = {"kimura","aoki","yamada","tanaka", "arai"};

```



```
    return 0;
}

int strorder( char s[], char t[] )
{
    省略
}

```

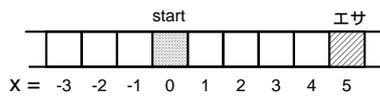
解答欄:

イ	
---	--

受験番号 _____ 氏名 _____

問題 4

1 次元の仮想的な世界に住むアリの動きをコンピュータでシミュレーションしたい．図のように 1 次元に並んだ矩形のセルを考え，セルに $x = \dots - 2, -1, 0, 1, 2 \dots$ と番号を付けておく．時刻 t において，あるセルに存在するアリは，次の時刻 $t + 1$ には，その他のアリの行動とは無関係に $1/3$ の確率で右隣のセルに移動し， $2/3$ の確率で左隣のセルに移動するものとする． $x = 5$ の位置にエサがあり，各アリは少なくとも時刻 $t = 10$ までにエサに到達しなければ死んでしまうものとする．今，時刻 $t = 0$ において中央のセル $x = 0$ にのみ 1000 匹のアリが存在したとして，時刻 $t = 10$ 以降に生きているアリの数を見積もるプログラムを作成せよ．ただし， $0 \leq r < 1$ の範囲の一様乱数 r を戻り値として返す double 型の関数 `rrand()` が利用できるとしてよい．もし以下の解答欄に書ききれない場合には，その旨を明記してこの解答用紙の裏面にわたっても良い．



解答欄：