

クラス	学籍番号	氏名

次のコードの実行結果（出力）を書け。

#include <stdio.h> #include <string.h>  int main(void) { char s[20] = { 'A', 'B', 'C', '\\0', 'E' }; printf("%zu\\n", (sizeof(s) / sizeof(s[0]))); printf("%zu\\n", strlen(s)); printf("%s\\n", s); s[3] = s[1]; printf("%zu\\n", strlen(s)); puts(s); }	20 3 ABC 5 ABCBE
--	------------------------------

#include <stdio.h>  void Draw(int n) { for (int y = 0; y < n; ++y) { for (int x = 0; x < n; ++x) { putchar((x <= y) ? 'A' : 'B'); }  putchar('\\n'); } }  int main(void) { Draw(6); }	ABBBBB AABBBB AAABBB AAAABB AAAAAB AAAAAA
---	--

#include <stdio.h> #include <stdbool.h>  bool A() { puts("A"); return true; }  bool B() { puts("B"); return false; }  int main(void) { if (A()) { if (A()) { puts("C"); } }  if (A() && (A()    B())) { puts("D"); } }
---

A
A
C
A
A
D

#include <stdio.h>  int main(void) { int a = 30, b = 40;  if (a < b) { int a = 80, b = 20; a += b; b = 50; }  printf("%d %d\\n", a, b); }
---

30 40
-------

#include <stdio.h>  typedef enum { RED, BLUE = 100, YELLOW, PINK = 99, GREEN, } Color;  int main(void) { printf("%d\\n", RED); printf("%d\\n", YELLOW); printf("%d\\n", GREEN); }
---

0
101
100

次の式の結果について、何型でどのような値になるかを書け。

	型 (int など)	値
3.0 + 1.0	double	4.0
floor(-3.9)	double	-4.0
(int)-3.9	int	-3
-3 < -2 < -1	int	0

次のコードを実行し、指定した入力（複数行）を一度に与えた場合の出力を書け。

#include <stdio.h>  int main(void) { char a, b, c; scanf("%c%c%c", &a, &b, &c);  printf("%c%c%c\\n", a, b, c); printf("%d\\n", (a + b + c)); }	入力 90 100 110 90 (空行) 115
---	---

#include <stdio.h>  int main(void) { int balance = 10000;  for (;;) { char action; int value; scanf(" %c %d", &action, &value);  if (value < 0) { puts("Invalid"); continue; }  if (action == 'q') { break; } else if (action == 'd') { balance += value; } else if (action == 'w') { if (balance < value) { puts("Insufficient"); continue; }  balance -= value; } else if (action == 'b') { printf("%d\\n", balance); } else { puts("Invalid"); } } }	入力 b 0 d 5000 d -2000 d 1000 b 0 w 20000 w 10000 b 0 d 5000 b 5 b 0 q 0 10000 Invalid 16000 Insufficient 6000 11000 11000
--	--

10進	16進	文字	10進	16進	文字	10進	16進	文字	10進	16進	文字
0	00	\\0 (ヌル文字)	32	20	space	64	40	@	96	60	`
1	01	SOH	33	21	!	65	41	A	97	61	a
2	02	STX	34	22	"	66	42	B	98	62	b
3	03	ETX	35	23	#	67	43	C	99	63	c
4	04	EOT	36	24	\$	68	44	D	100	64	d
5	05	ENQ	37	25	%	69	45	E	101	65	e
6	06	ACK	38	26	&	70	46	F	102	66	f
7	07	\\a (ベル)	39	27	'	71	47	G	103	67	g
8	08	\\b (バックスペース)	40	28	(	72	48	H	104	68	h
9	09	\\t (水平タブ)	41	29	)	73	49	I	105	69	i
10	0A	\\n (改行)	42	2A	*	74	4A	J	106	6A	j
11	0B	\\v (垂直タブ)	43	2B	+	75	4B	K	107	6B	k
12	0C	\\f (改ページ)	44	2C	,	76	4C	L	108	6C	l
13	0D	\\r (復帰)	45	2D	-	77	4D	M	109	6D	m
14	0E	SO	46	2E	.	78	4E	N	110	6E	n
15	0F	SI	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	48	30	0	80	50	P	112	70	p
17	11	DC1	49	31	1	81	51	Q	113	71	q
18	12	DC2	50	32	2	82	52	R	114	72	r
19	13	DC3	51	33	3	83	53	S	115	73	s
20	14	DC4	52	34	4	84	54	T	116	74	t
21	15	NAK	53	35	5	85	55	U	117	75	u
22	16	SYN	54	36	6	86	56	V	118	76	v
23	17	ETB	55	37	7	87	57	W	119	77	w
24	18	CAN	56	38	8	88	58	X	120	78	x
25	19	EM	57	39	9	89	59	Y	121	79	y
26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	59	3B	;	91	5B	[	123	7B	{
28	1C	FS	60	3C	<	92	5C	\\	124	7C	
29	1D	GS	61	3D	=	93	5D	]	125	7D	}
30	1E	RS	62	3E	>	94	5E	^	126	7E	~
31	1F	US	63	3F	?	95	5F	_	127	7F	DEL

(参考) ASCII 表

誤っている部分があれば下線を引き、その部分を置き換える正しい記述を書け。誤っている部分が無い場合は「正しい」と書け。

bool 型の変数 a に整数を代入したあと、a は true または false のどちらかである

正しい

ある関数 F が引数として構造体 S をとるとする。S のメンバ変数の個数を増やす際、関数 F のプロトタイプ宣言も同時に修正しないとコンパイルエラーになる

を修正する必要はない

sizeof は関数であり、講義で書いた COUNTOF は関数形式のマクロである

演算子

算術演算子 (+, -, \*, /, %) や比較演算子 (<, <=, >, >=, ==, !=) は左結合である。左結合の演算子が同じ優先順位で並んだ場合、左から右へ順に計算（評価）される

正しい

要約して本質的な機能を説明せよ。1 行ずつ説明した場合は不正解とする。

```
#include <stdio.h>
#include <ctype.h>
#include <stdbool.h>

int main(void)
{
    bool newWord = true;

    for (;;)
    {
        char c = (char)getchar();

        if (c == '\n')
        {
            break;
        }

        if (c == ' ')
        {
            newWord = true;
        }
        else
        {
            if (newWord && islower(c))
            {
                c = 'A' + (c - 'a');
            }

            newWord = false;
        }

        putchar(c);
    }

    putchar('\n');
}
```

標準入力から改行が入力されるまで文字列を読み込み、半角空白で区切られた各単語の先頭の文字を大文字に変更して出力する。

```
bool Check(const char s[], const char t[])
{
    int sLength = (int)strlen(s);
    int tLength = (int)strlen(t);

    if (sLength < tLength)
    {
        return false;
    }

    for (int i = 0; i < tLength; ++i)
    {
        if (s[sLength - tLength + i] != t[i])
        {
            return false;
        }
    }

    return true;
}
```

文字列 s が文字列 t で終わる場合 true を、それ以外の場合は false を返す。

就寝時刻と起床時刻を 24 時間表記で入力すると睡眠時間を出力するプログラムである。穴埋め部分のコードを書いて完成させよ。24 時間以上の睡眠時間は想定しなくてよい。8:15 は「8 15」、20:00 は「20 0」と入力する。

```
#include <stdio.h>

int main(void)
{
    int sh, sm; // 就寝時刻の時と分
    int eh, em; // 起床時刻の時と分
    scanf("%d %d %d %d", &sh, &sm, &eh, &em);

    printf("%d 時間 %d 分\n", (result / 60), (result % 60));
}
```

<入力例>  
23 30  
6 0

```
int start = sh * 60 + sm; // 分に変換
int end = eh * 60 + em; // 分に変換

if (end < start) // 起床時刻が就寝時刻より小さい場合
{
    end += 24 * 60; // 翌日の時刻に変換
}

int result = end - start; // 起床時刻から就寝時刻を引く
```

main 関数終了時点での配列 grid の状態（要素の並び）を書け。

```
#include <stdio.h>
#define W 6
#define H 5

void Flip(int grid[], int sx, int sy, int w, int h)
{
    for (int y = sy; y < (sy + h); ++y)
    {
        if ((y < 0) || (H <= y))
        {
            continue;
        }

        for (int x = sx; x < (sx + w); ++x)
        {
            if ((x < 0) || (W <= x))
            {
                continue;
            }

            grid[y * W + x] = !grid[y * W + x];
        }
    }
}
```

Flip は、グリッド上の矩形領域において 0 と 1 を反転する操作

```
int main(void)
{
    int grid[H * W] = { 0 };
    Flip(grid, 0, 0, 2, 2);
    Flip(grid, 2, 1, 3, 3);
    Flip(grid, 3, 0, 3, 3);
    Flip(grid, 0, 3, 1, 3);
    Flip(grid, 5, 4, 3, 3);
}
```

先頭	1	1	0	1	1	1	
	1	1	1	0	0	1	
	0	0	1	0	0	1	
	1	0	1	1	1	0	
	1	0	0	0	0	1	末尾