

1. printf() が表示するものを書いて表を完成しなさい。なお、空白は「□」と書き表すこと。(2点×10問=20点)

ヒント: $65_{10} = 41_{16}$

番号	printf()	出力
(a)	printf("%3d", 65);	□ 6 5
(b)	printf("%03d", 65);	0 6 5
(c)	printf("%-3d", 65);	6 5 □
(d)	printf("%x", 65);	4 1
(e)	printf("%c", 65);	A
(f)	printf("%3s", "ab");	□ a b
(g)	printf("%-3s", "ab");	a b □
(h)	printf("%%");	%
(i)	printf("\\n");	\\n
(j)	printf("\\\"abc\\");	" a b c "

2. 文字列が文字配列に格納された時、文字配列に格納される文字コードを先頭から順に16進数で書きなさい。文字列終端マーク(00H)も忘れず書くこと。(5点×2問=10点)

(1) "abc"

61H, 62H, 63H, 00H

(2) "ab\\\""

61H, 62H, 22H, 00H

参考: 文字コード表

		(上位3ビット)							
		0	1	2	3	4	5	6	7
(下位4ビット)	0	NUL	DLE	(SP)	0	@	P	`	p
	1	SOH	DC1	!	1	A	Q	a	q
	2	STX	DC2	"	2	B	R	b	r
	3	ETX	DC3	#	3	C	S	c	s
	4	EOT	DC4	\$	4	D	T	d	t
	5	ENQ	NAK	%	5	E	U	e	u
	6	ACK	SYN	&	6	F	V	f	v
	7	BEL	ETB	'	7	G	W	g	w
	8	BS	CAN	(8	H	X	h	x
	9	HT	EM)	9	I	Y	i	y
	A	LF	SUB	*	:	J	Z	j	z
	B	VT	ESC	+	;	K	[k	{
	C	FF	FS	,	<	L	\	l	
	D	CR	GS	-	=	M]	m	}
	E	SO	RS	.	>	N	^	n	~
	F	SI	US	/	?	O	_	o	DEL

3. 次のC言語プログラムの実行結果を答えなさい。(10点)

```
// ex2.c
#include <stdio.h>

void copy(char to[], char from[]) {
    int i = 0;
    while ((to[i]=from[i])!='\0')
        i++;
}

int main() {
    char buf[20];

    copy(buf, "abc");
    printf("[%s]\n", buf);

    copy(buf, "0123456789");
    buf[3] = 'C';
    printf("[%s]\n", buf);    // 何を表示するか?

    buf[5] = '\0';
    printf("[%s]\n", buf);    // 何を表示するか?

    return 0;
}
```

\$ ex2

[abc]

<--- 一つ目の printf の出力

[012C456789]

[012C4]

4. 次の C 言語プログラムの実行結果を答えなさい。(5 点)

```
// ex3.c
#include <stdio.h>
void f(int a) {
    a = 3 * a;
}

int main() {
    int a = 3;
    f(a);
    printf("%d\n", a);
    return 0;
}
```

\$ ex3

3

5. 次の C 言語プログラムの実行結果を答えなさい。(5 点)

```
// ex4.c
#include <stdio.h>
void g(int a[]) {
    a[0] = a[0] * 3;
    a[1] = a[1] * 3;
}

int main() {
    int a[2];
    a[0] = 1;
    a[1] = 2;
    g(a);
    printf("%d,%d\n",a[0],a[1]);
    return 0;
}
```

\$ ex4

3, 6

6. 次の C 言語プログラムについて答えなさい。

```
// ex5.c
#include <stdio.h>
#define N 9
int i;
void f(int j) {
    for (i=1; i<=N; i++)
        printf("%3d", i*j);
    printf("\n");
}

int main() {
    for (i=1; i<=N; i++)
        f(i);
    return 0;
}
```

(1) このプログラムの出力を答えなさい。(5 点)

\$ ex5

1 2 3 4 5 6 7 8 9

(2) このプログラムは以下のような九九の表を表示する予定でした。正しく動作するように上のプログラム中に修正を書き込みなさい。(5 点)

\$ ex5

1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

7. 実行結果を参考に、次の C 言語プログラムを完成しなさい。strFill 関数は、buf に c が len 個からなる文字列を作成する。文字列として完成するために文字列終端マークを付加することを忘れないように。(10 点)

```
// ex6.c
#include <stdio.h>
void strFill(char buf[], char c, int len) {

    for (int i=0; i<len; i++)
        buf[i] = c;
    buf[len] = '\0';

}

int main() {
    char buf[10];
    strFill(buf, 'a', 3);
    printf("%s\n", buf);
    strFill(buf, 'b', 4);
    printf("%s\n", buf);
    return 0;
}

/* 実行結果
$ ex6
aaa
bbbb
*/
```

8. 実行結果を参考に、次の C 言語プログラムを完成しなさい。strCat 関数は buf の文字列に str の文字列を追加する。なお、main 関数と実行結果は 6. の strFill 関数を使用した例になっている。(10 点)

```
// ex7.c
#include <stdio.h>
void strCat(char buf[], char str[]) {

    int i=0;
    while (buf[i]!='\0')
        i++;
    for (int j=0;
        str[j]!='\0'; j++) {
        buf[i]=str[j];
        i++;
    }
    buf[i]='\0';

}

int main() {
    char buf[10];
    strFill(buf, 'a', 3);
    strCat(buf, "bbb");
    printf("%s\n", buf);
    strCat(buf, "cc");
    printf("%s\n", buf);
    return 0;
}

/* 実行結果
$ ex7
aaabbb
aaabbbcc
*/
```

9. 実行結果を参考に、次の C 言語プログラムを完成しなさい。このプログラムは入力中の数字を全て '*' に置き換えて出力するものである。(10 点)

```
// ex9.c
#include <stdio.h>

int main() {

    int c;

    while ((c=getchar())!=EOF) {
        if ('0'<=c && c<='9')
            c = '*';
        putchar(c);
    }

    return 0;

}

/* 実行結果
$ ex9
abc0def9ghi          <--- 入力
abc*def*ghi           <--- 出力
jkl23mn4op           <--- 入力
jkl**mn*op           <--- 出力
^D                   <--- EOF 入力
$
*/
```

10. 実行結果を参考に、次の C 言語プログラムを完成しなさい。なお、innerProduct 関数は n 次元のベクトル a、b の内積を計算する。(10 点)

```
// exA.c
#include <stdio.h>

// n 次元のベクトル a, b の内積を計算する
double innerProduct(double a[],
                    double b[], int n) {

    double x = 0.0;
    for (int i=0; i<n; i++) {
        x = x + a[i] * b[i];
    }
    return x;

}

int main() {
    // 配列を作り初期化する
    double a[] = {1.0, 2.0, 3.0};
    double b[] = {1.0, 0.0, 1.0};

    double x = innerProduct(a,b,3);
    printf("%f\n", x);
    return 0;
}

/* 実行結果
$ exA
4.000000
*/
```