

1. 別紙のプログラム p1 について答えなさい。(3 点× 6 問=18 点)

(a) 実行結果を完成しなさい。

(a):	1
(b):	2
(c):	1

(b) 空欄に適切な漢字 1 文字を答えなさい。

C 言語の関数では、呼出し側の (a)

引数の値が関数の (b) 引数にコピーされるので、上のような実行結果になった。このような引数の渡し方は (c) 渡しと呼ばれる。

(a)	実
(b)	仮
(c)	値

2. 別紙のプログラム p2 について答えなさい。(3 点× 5 問=15 点)

実行結果を完成しなさい。但し、エラーになる行には「エラー」と書くこと。

(a):	2
(b):	3
(c):	5
(d):	エラー
(e):	1

3. 別紙のプログラム p3 について答えなさい。(3 点× 5 問=15 点)

(a) プログラム中の空欄に適切な記述を答えなさい。

(1)	int *a
(2)	&

(b) 実行結果を完成しなさい。

(a):	1
(b):	2
(c):	2

4. 別紙のプログラム p4 と別紙「p4 の実行例」をよく見て答えなさい。(3 点× 4 問=12 点)

(a) プログラム中の空欄に適切な記述を答えなさい。

(1)	fp==NULL
(2)	argc!=2

(b) 実行結果中の空欄に適切な記述を答えなさい。(改行コードに注意)

(a)	Usage: p4 <file>
(b)	61 62 63 0a

5. 別紙のプログラム p4 をもとに、コマンド行引数がある場合は、それをファイル名と見做し標準出力の代わりにファイルへ出力するプログラム p5 に改造します。(なお、余計な引数が無い時は標準出力に出力します。) 以下の問に答えなさい。(3 点× 3 問=9 点)

- (a) dumpFile() 関数中の全ての printf() 関数は、同じルールで書換える必要があります。
例えば printf("\n") はどのように書き換えるべきか答えなさい。

fprintf(dFp, "\n")

- (b) プログラム中の空欄に適切な記述を答えなさい。

(1) dFp = fopen(argv[2], "w");

(2) dFp = stdout;

6. 別紙のプログラム p6 について答えなさい。(3 点×6 問=18 点)

- (a) プログラム中の空欄に適切な記述を答えなさい。

(1) struct X *p

(2) struct X q

(a) (1 , 2)

(b) (3 , 4)

(c) (1 , 2)

(d) (1 , 2)

- (b) 関数 f() の出力を答えなさい。

7. 別紙のプログラム p7 について答えなさい。(3 点×2 問=6 点)

実行結果のようになるようにプログラム中の空欄に適切な記述を答えなさい。なお、(1) では sizeof() を用いること。

(1) sizeof(x)/sizeof(struct X)

(2) "(%d,%s)\n", x[i].a, x[i].s

8. 別紙のプログラム p8 の実行結果を答えなさい。(3 点)

\$ p8 1 2 3 4 5 6 7 8 9

1 2 3 4 5 6 7 8 9

9. 別紙のプログラム p9 の関数 delNode() を完成しなさい。

delNode() は引数に渡されたノードをリストから外す関数である。(4 点)

```
void delNode(Node *p) {
```

```
    p->prev->next = p->next;
```

```
    p->next->prev = p->prev;
```

```
}
```

別紙プログラム

```
// p1.c
#include <stdio.h>
void f(int a) {
    printf("(a):%d\n", a);
    a = 2;
    printf("(b):%d\n", a);
}
int main() {
    int a = 1;
    f(a);
    printf("(c):%d\n", a);
    return 0;
}
```

```
// p2.c
#include <stdio.h>
int a = 1;
void f() {
    printf("(e):%d\n", a);
}
int main() {
    int a = 2;
    for (int i=3; i<4; i++) {
        printf("(a):%d\n", a);
        printf("(b):%d\n", i);
        a = 5;
    }
    printf("(c):%d\n", a);
    printf("(d):%d\n", i);
    f();
    return 0;
}
```

```
// p3.c
#include <stdio.h>
void f(###(1)###) {
    printf("(a):%d\n", *a);
    *a = 2;
    printf("(b):%d\n", *a);
}
```

```
int main() {
    int a = 1;
    f(###(2)### a);
    printf("(c):%d\n", a);
    return 0;
}
```

```
// p4.c
#include <stdio.h>
#include <stdlib.h>
void usage(char *prog) {
    fprintf(stderr,
        "Usage: %s <file>\n", prog);
    exit(1);
}
FILE *eOpen(char *fname, char *mode){
    FILE *fp=fopen(fname,mode);
    if (###(1)###){
        perror(fname);
        exit(1);
    }
    return fp;
}
void dumpFile(FILE *fp) {
    int ch, i=0;
    while ((ch=fgetc(fp))!=EOF) {
        if (i%16==0) printf("%08x:", i);
        printf("_%02x", ch);
        if (i%16==15) printf("\n");
        i++;
    }
    if (i%16!=0) printf("\n");
}
int main(int argc, char *argv[]) {
    FILE *sFp;
    if (###(2)###) usage(argv[0]);
    sFp = eOpen(argv[1], "rb");
    dumpFile(sFp);
    return 0;
}
```

```
// p4 の実行例
$ p4
###(a)###
$ cat a.txt
abc
$ cat b.txt
cat: b.txt: No such file or directory
$ p4 a.txt
00000000 : ###(b)###
$ p4 b.txt
b.txt:No such file or directory
$
```

```
// p5.c
#include <stdio.h>
#include <stdlib.h>
void usage(char *prog) {
    fprintf(stderr,
        "Usage: %s <file1> <file2>\n",
        prog);
    exit(1);
}
FILE *eOpen(char *fname, char *mode){
    ... 省略...
}
void dumpFile(FILE *sFp, FILE *dFp){
    ... 省略...
}
int main(int argc, char *argv[]) {
    FILE *sFp, *dFp;
    if (argc<2 || 3<argc)
        usage(argv[0]);
    sFp = eOpen(argv[1], "rb");
    if (argc==3) ###(1)###
    else ###(2)###
        dumpFile(sFp, dFp);
    return 0;
}
```

```
// p6.c
#include <stdio.h>
struct X { int a; int b; };
struct X x = {1, 2};
struct X y = {3, 4};
void f(###(1)###) {
    printf("(%d,%d)\n", p->a, p->b);
}
void g(###(2)###) {
    q.a = 0;
    q.b = 0;
}
int main() {
    f( &x );    // (a)
    f( &y );    // (b)
    y = x;
    f( &y );    // (c)
    g(x);
    f( &x );    // (d)
    return 0;
}
```

```
// p7.c
#include <stdio.h>
struct X { int a; char *s; };
struct X x[] =
    {{1, "abc"}, {2, "def"}};
int main() {
    for (int i=0; i<###(1)###; i++)
        printf(###(2)###);
    return 0;
}

/* 実行結果
$ p7
(1,abc)
(2,def)
*/
```

```
// p8.c
#include <stdio.h>
#include <stdlib.h>
struct N {
    int val;
    struct N *prev; struct N *next;
};
typedef struct N Node;
Node *list;

void insNode(Node *p, Node *q) {
    q->prev = p->prev;
    q->prev->next = q;
    q->next = p;
    p->prev = q;
}

int main(int argc, char *argv[]) {
    Node *p;
    list = malloc(sizeof(Node));
    list->next = list->prev = list;

    for (int i=1; i<argc; i++) {
        int x = atoi(argv[i]);
        p = malloc(sizeof(Node));
        p->val = x;
        insNode(list, p);
    }

    for (p=list->next;p!=list;p=p->next)
        printf("%d", p->val);
    printf("\n");
    return 0;
}
```

```
// p9.c
... 前の方は p8 と同じ ...

void delNode(Node *p) {
    // ここを完成する(2行で書ける)
}

int main(int argc, char *argv[]) {
    Node *p;
    list = malloc(sizeof(Node));
    list->next = list->prev = list;

    for (int i=1; i<argc; i++) {
        ... p8 と同じ ...
    }

    q = ... // どれかのノードを選び
    delNode(q); // リストから外す
    free(q); // 領域を解放する

    for (p=list->next;p!=list;p=p->next)
        printf("%d", p->val);
    printf("\n");
    return 0;
}
```

(上位3ビット)

	0	1	2	3	4	5	6	7
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	

(下位4ビット)

参考：ASCII コード表