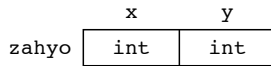


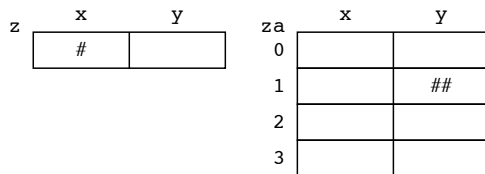
1. 問題を読み適切な C 言語プログラムの一部を書きなさい。

(1) 図のような構造を持つ構造体型 zahyo の宣言 (6 点)



```
struct zahyo {  
    int x;  
    int y;  
};
```

(2) (1) を利用して、図の変数 z と配列変数 za を宣言 (6 点)

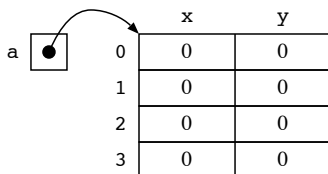


```
struct zahyo z;  
struct zahyo za[4];
```

(3) (2) の図中 '#' と '##' の位置に 100 と 200 を代入 (6 点)

```
z.x = 100;  
za[1].y = 200;
```

(4) (1) を利用して、図のようなポインタ変数 a を宣言 (3 点)



```
struct zahyo *a;
```

(5) (4) の図のような領域を malloc を用いて a に割付け (3 点)

```
a=malloc(sizeof(struct zahyo)*4);
```

(6) (5) で割付けた領域を図のように 0 でクリア (3 点)

```
int i;  
  
for (i=0; i<4; i++) {  
    a[i].x = 0;  
    a[i].y = 0;  
}
```

(7) zahyo 構造体 n 個からなる配列領域を割付け、内容を 0 でクリアし、領域へのポインタを返す allocZahyo 関数 (3 点) (なお、エラー処理は省略して良いものとする。)

```
struct zahyo *allocZahyo(int n) {  
  
    int i;  
    struct zahyo *a;  
    a = malloc(sizeof(  
                                struct zahyo)*n);  
    for (i=0; i<n; i++) {  
        a[i].x = 0;  
        a[i].y = 0;  
    }  
    return a;  
}
```

2. 次のプログラムは、コマンド行の全ファイルについて内容を表示するものです。実行例を参考に空欄を埋めなさい。
(3 点× 10 問=30 点)

```
// ex1.c
#include <stdio.h>
#include <stdlib.h>

main(int argc, char *argv[]) {
    int i;
    /* (A) */ ch;
    /* (B) */ fp;
    for (i=/* (C) */; i<argc; i++) {
        fp = fopen(/* (D) */, /* (E) */);
        if (/* (F) */) {
            fprintf(/* (G) */,
                    "can't open '%s'\n",
                    /* (H) */);
            /* (I) */;
        }
        while((ch=getc(fp))!=EOF)
            putchar(ch);

        /* (J) */;
    }
    exit(0);
}

/* 実行例
$ ex1 a.txt
aaaaa
aaaaa
$ ex1 b.txt
bbbbbb
bbbbbb
$ ex1 a.txt b.txt
aaaaa
aaaaa
bbbbbb
bbbbbb
$ ex1 c.txt
can't open 'c.txt'
$
*/
```

(A)	int
(B)	FILE *
(C)	1
(D)	argv[i]
(E)	"r"
(F)	fp == NULL
(G)	stderr
(H)	argv[i]
(I)	exit(1)
(J)	fclose(fp)

3. プログラムを読み実行例の空欄を埋めなさい。(10 点)

```
// ex2.c
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <unistd.h>

#define N 100

main() {
    char buf[N];
    int fd = open("ex2.c",O_RDONLY);

    for (;;) {
        int l = read(fd, buf, N);
        printf("%d\n", l);
        if (l==0) break;
    }

    exit(0);
}

/* 実行例
$ ls -l ex2.c
-rw-r--r-- 1 sigemura kan 372 ... ex2.c
$ ex2
??????????? ← 何出力されるか答える
????
$
*/

... : は時刻の表示を省略
```

100
100
100
72
0

4. 次のようなファイルがあります。sigemura、momota は kan グループのユーザ、i11abc は gak グループのユーザとします。(6 点×2 問=12 点)

```
$ ls -l
-rwxr-x-wx 1 sigemura kan 100 ... a.txt
-rwxr-xr-x 1 i11abc  gak  15 ... b.txt
-rw--w---- 1 momota  kan 321 ... c.txt
$

... : は時刻の表示を省略
```

(1) 各ファイルのモードを 8 進数で答えなさい。

a.txt	753
b.txt	755
c.txt	620

(2) ユーザが読み出せるファイルの名前を全て答えなさい。

sigemura	a.txt, b.txt
i11abe	b.txt
momota	a.txt, b.txt, c.txt

5. ファイルを書き込み用に関く open システムコールの第 2 以下の引数の記述を答えなさい。(4 点×2 問=8 点)

```
fd = open("a.txt", /* ここに何を書くか */);
```

(1) ファイルが存在しない場合は rw-r-r- で作成する。ファイルが存在する場合は内容を、一旦、消去する。

O_WRONLY | O_CREAT | O_TRUNC, 0644

(2) ファイルが存在する場合は、内容を残しファイルの最後から追記するモードにする。

O_WRONLY | O_APPEND

6. 実行例のように、コマンド行引数を逆順にカンマ区切りで出力するプログラム (ex3.c) を書きなさい。(10 点)

```
$ ex3  
  
$ ex3 abc  
abc  
$ ex3 abc def ghi  
ghi,def,abc  
$
```

```
// ex3.c  
#include <stdio.h>  
#include <stdlib.h>  
  
main(int argc, char *argv[]) {  
    int i;  
  
    for (i=argc-1; i>=1; i--) {  
        printf("%s", argv[i]);  
        if (i>1) printf(",");  
    }  
    printf("\n");  
    exit(0);  
}
```