

1. 以下のプログラム (p1.c～p6.c) を読み下の問題に答えなさい。(3点×15問=45点)

```
/* p1.c */
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <unistd.h>
#define BSIZ 1024
void err_exit(char *s) {
    perror( s );
    exit(1);
}
int main(int argc, char *argv[]) {
    int fd1, fd2;
    int len;
    char buf[BSIZ];
    if (argc!=3) {
        fprintf(stderr,
            "Usage : %s source target\n", argv[0]);
        exit(1);
    }
    fd1 = open(argv[1], O_RDONLY);
    if (fd1<0) err_exit( argv[1] );
    fd2 = open(argv[2],
        O_WRONLY|O_CREAT|O_TRUNC,
        0644);
    if (fd2<0) err_exit( argv[2] );
    while ((len=read(fd1, buf,BSIZ))>0) {
        write(fd2,buf,len);
    }
    close(fd1);
    close(fd2);
    exit(0);
}
```

```
/* p2.c */
#include <stdio.h>
#include <stdlib.h>
#include <sys/stat.h>
int main(int argc, char *argv[]) {
    int i;
    int err = 0;
    if (argc<2) {
        fprintf(stderr,
            "Usage: %s directory ...\n", argv[0]);
        exit(1);
    }
    for (i=1; i<argc; i++) {
        if (mkdir(argv[i],0755)<0) {
            perror(argv[i]);
            err = 1;
        }
    }
    exit(err);
}
```

```
/* p3.c */
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(int argc, char *argv[]) {
    int i, err = 0;
    if (argc<2) {
        fprintf(stderr,
            "Usage: %s file ...\n", argv[0]);
        exit(1);
    }
    for (i=1; i<argc; i++) {
        if (unlink(argv[i])<0) {
            perror(argv[i]);
            err = 1;
        }
    }
    exit(err);
}
```

```
/* p4.c */
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
int main(int argc, char *argv[]) {
    if (argc!=3) {
        fprintf(stderr,
            "Usage: %s file1 file2\n", argv[0]);
        exit(1);
    }
    if (link(argv[1], argv[2])<0) {
        perror(argv[0]);
        exit(1);
    }
    exit(0);
}
```

```
/* p5.c */
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
int main(int argc, char *argv[]) {
    if (argc!=3) {
        fprintf(stderr,
            "Usage: %s file1 file2\n", argv[0]);
        exit(1);
    }
    if (symlink(argv[1], argv[2])<0) {
        perror(argv[2]);
        exit(1);
    }
    exit(0);
}
```

```
/* p6.c */
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
    if (argc!=3) {
        fprintf(stderr,
            "Usage: %s from to\n", argv[0]);
        exit(1);
    }
    if (rename(argv[1], argv[2])<0) {
        perror(argv[0]);
        exit(1);
    }
    exit(0);
}
```

実行結果：

```
$ p2 D1 D2
$ echo aaa > D1/a.txt
$ p1
[A]
$ p1 D1/a.txt D1/b.txt
$ ls -l D1
-rw-r--r-- [B] sigemura staff 4 Jun 6 11:02 a.txt
-rw-r--r-- [C] sigemura staff 4 Jun 6 11:02 b.txt
$ p4 D1/a.txt D1/c.txt
$ ls -l D1
-rw-r--r-- [D] sigemura staff 4 Jun 6 11:02 a.txt
-rw-r--r-- [E] sigemura staff 4 Jun 6 11:02 b.txt
-rw-r--r-- [F] sigemura staff 4 Jun 6 11:02 c.txt
$ p6 D1/a.txt D2/a.txt
$ ls -l D1
-rw-r--r-- [G] sigemura staff 4 Jun 6 11:02 b.txt
-rw-r--r-- [H] sigemura staff 4 Jun 6 11:02 c.txt
$ p3 D2/a.txt
$ ls -l D1
-rw-r--r-- [I] sigemura staff 4 Jun 6 11:02 b.txt
-rw-r--r-- [J] sigemura staff 4 Jun 6 11:02 c.txt
$ p3 D1/b.txt
$ ls -l D1
-rw-r--r-- 1 sigemura staff 4 Jun 6 11:02 c.txt
$ p1 D2/a.txt D1/c.txt
[K]
$ p5 [L] [M]
$ cat D2/a.txt
aaa
$ p3 D1/c.txt
$ p5 [N] [O]
$ cat D2/a.txt
cat: D2/a.txt: Too many levels of symbolic links
$
```

問題：次の表に、実行結果の空欄 ([A] 等) の内容を答えなさい。

[A]	Usage : p1 source target	
[B]	1	[C] 1
[D]	2	[E] 1
[F]	2	[G] 1
[H]	2	[I] 1
[J]	1	
[K]	D2/a.txt: No such file or directory	
[L]	../D1/c.txt	
[M]	D2/a.txt	
[N]	../D2/a.txt	
[O]	D1/c.txt	

2. シグナル処理に関する問題に答えなさい。

(1) 次のプログラム (ex1.c) の実行中に Ctrl-C を何度か入力します。そのときの動作 (表示されるものや、プログラムの停止・終了等) を説明しなさい。

```
#include <stdio.h>
#include <signal.h>
#include <stdlib.h>
#include <unistd.h>
void handler3(int s) {
    signal(SIGINT, SIG_DFL);
    printf("\nKONICHIWA\n");
}
void handler2(int s) {
    signal(SIGINT, handler3);
    printf("\nANIHASEYO\n");
}
void handler1(int s) {
    signal(SIGINT, handler2);
    printf("\nHELLO\n");
}
int main() {
    signal(SIGINT, handler1);
    for (;;) {
        printf("OYASUMI\n");
        pause();
    }
    exit(1);
}
```

a. プログラム起動時の動作を説明しなさい。(5点)

プログラムは起動されると”OYASUMI”と表示する。その後、プログラムは停止する。

b. Ctrl-C を何度か入力したときの動作を説明しなさい。
(表示されるものは全て書くこと。プログラムが終了するまでの過程を説明すること。)(10点)

1度目に Ctrl-C 入力したとき、プログラムは”HELLO”と表示する。その後、プログラムは”OYASUMI”と表示して停止する。

2度目に Ctrl-C 入力したとき、プログラムは”ANIHASEYO”と表示する。その後、プログラムは”OYASUMI”と表示して停止する。

3度目に Ctrl-C 入力したとき、プログラムは”KONICHIWA”と表示する。その後、プログラムは”OYASUMI”と表示して停止する。

4度目に Ctrl-C 入力したとき、プログラムは終了する。

(2) 次のプログラム (ex2.c) の動作について説明しなさい。

```
#include <stdio.h>
#include <signal.h>
#include <stdlib.h>
#include <unistd.h>
void inthandler(int s) {
    signal(SIGINT , SIG_DFL);
    alarm(5);
    printf("\nINT\n");
}
void alarmhandler(int s) {
    signal(SIGINT , inthandler);
    printf("ALRM\n");
}
int main() {
    signal(SIGINT , inthandler);
    signal(SIGALRM , alarmhandler);
    for (;;) {
        printf("PAUSE\n");
        pause();
    }
    exit(1);
}
```

a. プログラム起動後 Ctrl-C を2回連続して入力しました。プログラム起動後に表示されるものを全て書きなさい。
(もしもプログラムが終了してプロンプト (\$) が表示されるなら、プロンプトも書くこと。また Ctrl-C をいつ入力したか分かるように書くこと)(5点)

PAUSE

^C

INT

PAUSE

^C

\$

b. プログラムを起動します。1秒待って Ctrl-C を入力しました。その後、10秒待って Ctrl-C を入力しました。プログラム起動後に表示されるものを全て書きなさい。
(Ctrl-C をいつ入力したか分かるように書くこと)(10点)

PAUSE

^C

INT

PAUSE

ALRM

PAUSE

^C

INT

PAUSE

ALRM (10秒後)

PAUSE

3. 次のプログラム (ex3.c) を読んで問に答えなさい。

```
#include <stdio.h>
#include <unistd.h>
#include <signal.h>
int main(){
    for(;;){
        sleep(5);
        printf("Hello!\n");
    }
}
```

(1) ex3.c は何をするか簡単に説明しなさい。(5点)

“Hello!” と5秒に一度表示するプログラム。Ctrl-C を押すまで終了しない。

(2) ex3.c と同じ動作をするプログラム (ex4.c) を sleep システムコール (関数) を使用しないで書きなさい。(10 点)
(#include は省略して良い。)

//sleep を使用せず、5 秒に一回、"Hello!" と表示

```
#include <stdio.h>
#include <unistd.h>
#include <signal.h>

void handler(){
}

int main(){
    signal(SIGALRM,handler);
    for(;;){
        alarm(5);
        pause();
        printf("Hello!\n");
    }
}
```

4. read システムコールを使用して、ファイルのサイズを求めるプログラム (ex5.c) を作りなさい。(10 点)
(#include とエラー処理は省略して良い。)

実行例: \$ ex5 ex5.c
ex5.c のサイズ=374 バイト
\$

```
/* ex5.c */
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <unistd.h>

#define BSIZ 1024

int main(int argc, char *argv[]) {
    int fd;
    int len, sum = 0;
    char buf[BSIZ];

    fd = open(argv[1], O_RDONLY);

    while ((len=read(fd, buf, BSIZ))>0) {
        sum += len;
    }

    printf("%s のサイズ=%d バイト\n", argv[1], sum);
    close(fd);
    exit(0);
}
```