システムプログラミング H24年度 学年末試験 (2013.02.07 重村 哲至)

IE3 ____番 氏名 _____

1. 配列データを扱うプログラムを書きなさい.

(1) int 型の大きさ 100 の配列領域を確保する malloc を完成しなさい. (5 点)

```
int *a = malloc(
          sizeof(int)*100 );
```

(2) struct complex 型 (問題 5 参照) の大きさ 100 の配列領域を確保する malloc を完成しなさい. (5 点)

```
struct complex *b = malloc(
sizeof(struct complex)*100);
```

(3) (1) で確保した領域を for 文を用いてクリアするプログラムを完成しなさい. (5 点)

```
int i;
for ( i=0; i<100; i++)
  a[i] = 0;
```

(4) (1) で確保した領域を bzero 関数を用いてクリアするプログラムを完成しなさい. (5 点)

```
bzero(a, sizeof(int) * 100);
```

(5) (2) で確保した領域を for 文を用いてクリアするプログラムを完成しなさい. (5 点)

```
int i;
for ( i=0; i<100; i++) {
  b[i].r = 0.0;
  b[i].i = 0.0;
}
```

(6) (2) で確保した領域を bzero 関数を用いてクリアするプログラムを完成しなさい. (5 点)

```
bzero(b,
  sizeof(struct complex)
                      * 100);
```

2. 次のプログラムをよく読んで問に答えなさい.

```
/* ex2.c */
#include <stdio.h>
#include <stdlib.h>
main(int argc, char *argv[]) {
  int i;
  for (i=0; i<argc; i++)
    printf("%s\n", argv[i]);
  exit(0);
}
```

(1) プログラム (ex2) の実行結果を完成しなさい. (5 点)

```
$ ex2 abc def
ex2
abc
def
```

(2) 実行結果に合わせて ex2.c を書き変えなさい. (5 点)

```
$ ex2 abc def
def
abc
```

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

システムプログラミング H24年度 学年末試験 (2013.02.07 重村 哲至)

IE3 ____番 氏名 _____

3. 次のプログラム (ex3.c) を読んで問に答えなさい.

```
 1: /* ex3.c */
 2: #include <stdio.h>
 3: #include <stdlib.h>
 4:
 5: main(int argc, char *argv[]) {
 6:  FILE *fp1 = stdin;
 7:  FILE *fp2 = stdout;
 8:  int  ch;
 9:
10:  if (argc>=2) fp1 = fopen(argv[1], "r");
11:  if (argc>=3) fp2 = fopen(argv[2], "w");
12:
13:  while ((ch=getc(fp1))!=EOF)
14:    putc(ch, fp2);
15:
16:  fclose(fp1);
17:  fclose(fp2);
18:
19:  exit(0);
20: }
```

(1) ex3 はコマンド行引数が無いとき何をするか簡単に説明しなさい. (5 点)

標準入力ストリームからの入力を，そのまま，標準出力ストリームに出力する. (通常はキーボード入力をそのままディスプレイに表示する.)

(2) ex3 はコマンド行引数が一つのとき何をするか簡単に説明しなさい. (5 点)

コマンド行引数で指定されたファイルの内容を，標準出力ストリームに出力する. (通常はファイルの内容をディスプレイに表示する.)

(3) ex3 はコマンド行引数が二つのとき何をするか簡単に説明しなさい. (5 点)

コマンド行の一つ目の引数で指定されたファイルの内容を，コマンド行の二つ目の引数で指定されたファイルへ書き込む. (一つ目のファイルの内容を 2 つ目のファイルにコピーする.)

(4) ex3.c に入力ファイルのオープンに失敗した際のエラー処理を追加します. どの行の後に追加するか明示した上で，追加するプログラムを書きなさい. (5 点)

１０行の次に以下を追加する.

```
if (fp1==NULL) {
  perror(argv[1]);
  exit(1);
}
```

4. 次のファイルをコピーするプログラム (ex4) について答えなさい.

```
/* ex4.c */
#include <stdlib.h>
#include <fcntl.h>
#include <unistd.h>

#define  LEN 100

main(int argc, char *argv[]) {
  int fd1, fd2;
  int len;
  char buf[LEN];

  fd1 = open(argv[1], O_RDONLY);
  fd2 = open(argv[2], /*(A)*/, 0644);

  while ((len=read(fd1,buf,/*(B)*/)>0) {
    write(fd2, buf, /*(C)*/ );
  }

  close(fd1);
  close(fd2);
  exit(0);
}
```

システムプログラミング H24年度 学年末試験 (2013.02.07 重村 哲至)

IE3 ____番 氏名 _____

(1) プログラム中の空欄(A),(B),(C)に適切な記述を書きなさい．(5点×3問=15点)

(A)	O_WRONLY|O_CREAT|O_TRUNC
(B)	LEN
(C)	len

(2) 550バイトのファイルをコピーする場合，ループが何度実行されるか答えなさい．(2点)

6 回

(3) プログラム中のLENの値によって，プログラムの性能がどのように変化するか，理由も含めて簡単に説明しなさい．(8点)

システムコールの呼出しは重い処理である．呼出した回数に，ほぼ，比例して実行時間が長くなる．同じサイズのファイルをコピーするとき．LENの値が小さいとループの繰り返し回数が多くなり，read, write システムコールを呼出す回数が多くなる．そのため，実行時間が余計にかかるようになりプログラムの性能が悪くなる．逆にLENの値が大きいとシステムコールを呼出す回数が少なくなり，性能が良くなる．

5. 複素数データを扱うプログラムを次のページに書きなさい．

複素数データ

```
struct complex {
  double r;       // 実数部
  double i;       // 虚数部
};
```

(1) 例のように使用する newComplex 関数．(5点)

例：実数部を10.5，虚数部を22.3で初期化した複素数データを作る．

```
struct complex *p;
p = newComplex(10.5, 22.3);
```

(2) 複素数の和を計算する addComplex 関数．(5点)

例：a,b の和を a に求める．

```
struct complex *a, *b;
a = newComplex(10.5, 22.3);
b = newComplex(5.5,  10.2);
addComplex(a, b);
```

実行後は a の内容が(16.0, 32.5)になる．

(3) 次のように使用できる addComplex2 関数．(5点)

例：a,b,c の和を a に求める．

```
struct complex *a, *b, *c;
a = newComplex(10.5, 22.3);
b = newComplex(5.5,  10.2);
c = newComplex(1.2,  1.0);
addComplex(addComplex2(a, b),c);
```

実行後は a の内容が(17.2, 33.5)になる．

システムプログラミング H24年度 学年末試験 (2013.02.07 重村 哲至)

IE3 ____番 氏名 _____

(1) newComplex 関数

```
struct complex *newComplex(double r, double i) {
  struct complex *t;
  t = malloc(sizeof(struct complex));
  t->r = r;
  t->i= i;
  return t;
}
```

(2) addComplex 関数 (void 型)

```
void addComplex(struct complex *x,
                struct complex *y) {
  x->r += y->r;
  x->i += y->i;
}
```

(3) addComplex2 関数

```
struct complex *addComplex2(struct complex *x,
                            struct complex *y) {
  x->r += y->r;
  x->i += y->i;
  return x;
}
```