

```
p1.cpp
#include <stdio.h>

void swap(int, int);

main()
{
    int a = 1;
    int b = 3;

    swap(a, b);
    printf("a = %d\nb = %d\n", a, b);
}

void swap(int x, int y)
{
    int temp;

    temp = x;
    x = y;
    y = temp;
}
```

```
p2.cpp
#include <stdio.h>

void swap(int*, int*);

main()
{
    int a = 1;
    int b = 3;

    swap(&a, &b);
    printf("a = %d\nb = %d\n", a, b);
}

void swap(int *x, int *y)
{
    int temp;

    temp = *x;
    *x = *y;
    *y = temp;
}
```

```

p3.cpp
#include <stdio.h>

main()
{
    int data[] = { 9, 8, 7, 6, 5, 4, 3, 2, 1, -1 };
    int *p, s = 0;

    p = data;
    while( *p != -1 ) {
        s += *p;
        p++;
    }
    printf("sum = %d\n", s);
}

```

```

p4.cpp
#include <stdio.h>

main()
{
    int *p, a = 1, i;

    p = &a;

    for(i = 0; i < 10; i++) {
        printf("%0x\n", p);
        p++;
    }
}

```

```

p5.cpp
#include <stdio.h>

main()
{
    double *p, a = 1.0;
    int i;

    p = &a;

    for(i = 0; i < 10; i++) {
        printf("%0x\n", p);
        p++;
    }
}

```

```

p6.cpp
#include <stdio.h>

main()
{
    char *p, a = 'a';
    int i;

    p = &a;

    for(i = 0; i < 10; i++) {
        printf("%0x\n", p);
        p++;
    }
}

```

```

p7.cpp
#include <stdio.h>

char *mystrcat(char *, char *);

main()
{
    char a[16] = "ABCDEFGH";
    char b[16] = "IJKLMNOP";

    printf("%s\n", mystrcat(a,b));
}

char *mystrcat(char *s1, char *s2)
{
    char *p;
    char *q;

    p = s1;
    while(*p != '\0')
        p++;

    q = s2;
    while(*q != '\0')
        *p++ = *q++;
    *p = '\0';

    return s1;
}

```

```

p8.cpp
#include <stdio.h>

char *mystrchr(const char *, int);

main()
{
    char *a = "ABCDEFGH";

    printf("%s\n", mystrchr(a, 'd'));
    printf("%d\n", mystrchr(a, 'd') - a + 1);
}

char *mystrchr(const char *s, int c)
{
    const char *p = s;

    while(*p != '\0') {
        if(*p == c)
            break;
        p++;
    }

    return *p == '\0' ? NULL : (char *)p;
}

```

```

p9.cpp
#include <stdio.h>

char *mystrcpy(char *, const char *);

main()
{
    char a[20] = "abcdef";

    printf("%s\n", mystrcpy(a, "wxyz"));
}

char *mystrcpy(char *s1, const char *s2)
{
    char *p = s1;
    const char *q = s2;

    while(*q != '\0')
        *p++ = *q++;
    *p = '\0';

    return s1;
}

```

```

p10.cpp
#include <stdio.h>

void maxmin(const int, const double [], double *, double *);

main()
{
    int n = 5;
    double x[] = { 2.3, 5.4, 1.7, 8.4, 5.9 };
    double min, max;

    maxmin(n, x, &min, &max);
    printf("min = %4.1f   max = %4.1f\n", min, max);
}

void maxmin(const int n, const double x[], double *min, double *max)
{
    int i;
    *min = x[0];
    *max = x[0];

    for(i=0; i<n; i++) {
        if(x[i] > *max)
            *max = x[i];
        if(x[i] < *min)
            *min = x[i];
    }
}

```

```

p11.cpp
#include <stdio.h>
typedef struct {
    double x;
    double y;
} complex;
complex cadd(complex, complex);
complex csub(complex, complex);
complex cmul(complex, complex);
complex cdiv(complex, complex);

main()
{
    complex a = { 2.0, 1.0 }, b = { 2.0, -1.0 }, c;

    c = cadd(a, b);
    printf("%6.2f + %6.2f i + %6.2f + %6.2f i = %6.2f + %6.2f i\n",
           a.x, a.y, b.x, b.y, c.x, c.y);
    c = csub(a, b);
    printf("%6.2f + %6.2f i - %6.2f + %6.2f i = %6.2f + %6.2f i\n",
           a.x, a.y, b.x, b.y, c.x, c.y);
    c = cmul(a, b);
    printf("%6.2f + %6.2f i * %6.2f + %6.2f i = %6.2f + %6.2f i\n",
           a.x, a.y, b.x, b.y, c.x, c.y);
    c = cdiv(a, b);
    printf("%6.2f + %6.2f i / %6.2f + %6.2f i = %6.2f + %6.2f i\n",
           a.x, a.y, b.x, b.y, c.x, c.y);
}

complex cadd(complex a, complex b)
{
    complex c;

    c.x = a.x + b.x;
    c.y = a.y + b.y;

    return c;
}

complex csub(complex a, complex b)
{
    complex c;

    c.x = a.x - b.x;
    c.y = a.y - b.y;

    return c;
}

complex cmul(complex a, complex b)
{
    complex c;

    c.x = a.x * b.x - a.y * b.y;
    c.y = a.x * b.y + a.y * b.x;

    return c;
}

complex cdiv(complex a, complex b)
{
    complex c;

    c.x = ( a.x * b.x + a.y * b.y ) / ( b.x * b.x + b.y * b.y );
    c.y = ( a.y * b.x - a.x * b.y ) / ( b.x * b.x + b.y * b.y );
}

```

```

    return c;
}
p12.cpp
#include <stdio.h>

typedef struct {
    double x;
    double y;
} complex;

void cadd(const complex*, const complex*, complex*);
void csub(const complex*, const complex*, complex*);
void cmul(const complex*, const complex*, complex*);
void cdiv(const complex*, const complex*, complex*);

main()
{
    complex a = { 2.0, 1.0 }, b = { 2.0, -1.0 }, c;

    cadd(&a, &b, &c);
    printf("%6.2f + %6.2f i + %6.2f + %6.2f i = %6.2f + %6.2f i\n",
           a.x, a.y, b.x, b.y, c.x, c.y);
    csub(&a, &b, &c);
    printf("%6.2f + %6.2f i - %6.2f + %6.2f i = %6.2f + %6.2f i\n",
           a.x, a.y, b.x, b.y, c.x, c.y);
    cmul(&a, &b, &c);
    printf("%6.2f + %6.2f i * %6.2f + %6.2f i = %6.2f + %6.2f i\n",
           a.x, a.y, b.x, b.y, c.x, c.y);
    cdiv(&a, &b, &c);
    printf("%6.2f + %6.2f i / %6.2f + %6.2f i = %6.2f + %6.2f i\n",
           a.x, a.y, b.x, b.y, c.x, c.y);
}

void cadd(const complex *a, const complex *b, complex *c)
{
    c->x = a->x + b->x;
    c->y = a->y + b->y;
}

void csub(const complex *a, const complex *b, complex *c)
{
    c->x = a->x - b->x;
    c->y = a->y - b->y;
}

void cmul(const complex *a, const complex *b, complex *c)
{
    c->x = a->x * b->x - a->y * b->y;
    c->y = a->x * b->y + a->y * b->x;
}

void cdiv(const complex *a, const complex *b, complex *c)
{
    c->x = ( a->x * b->x + a->y * b->y ) / ( b->x * b->x + b->y * b->y );
    c->y = ( a->y * b->x - a->x * b->y ) / ( b->x * b->x + b->y * b->y );
}

```

```

p13.cpp
#include <stdio.h>
#include <string.h>

#define SIZE 100

main(int argc, char *argv[])
{
    FILE *fp;
    char buf[SIZE + 1];
    int flag, i;

    if(argc == 3 && strcmp(argv[1], "-n") == 0) {
        flag = 1;
        fp = fopen(argv[2], "r");
    }else {
        flag = 0;
        fp = fopen(argv[1], "r");
    }
    i = 0;
    while(fgets(buf, SIZE, fp) != NULL){
        i++;
        if(flag == 1)
            printf("%06d : ", i);
        printf("%s", buf);
    }
    fclose(fp);
}

```

```

p14.cpp
#include <stdio.h>
#include <stdlib.h>

#define SIZE 100

main(int argc, char *argv[])
{
    FILE *fp;
    char buf[SIZE + 1];
    int flag, i, line;

    if(argc == 3) {
        line = atoi(argv[1]);
        fp = fopen(argv[2], "r");
    }else {
        line = 5;
        fp = fopen(argv[1], "r");
    }
    for(i = 0; i < line; i++) {
        fgets(buf, SIZE, fp);
        printf("%s", buf);
    }
    fclose(fp);
}

```

```

p15.cpp
#include <stdio.h>
#include <stdlib.h>

#define SIZE 100

main(int argc, char *argv[])
{
    FILE *fp;
    char buf[SIZE + 1];
    int flag, i, line, last;

    if(argc == 3) {
        line = atoi(argv[1]);
        fp = fopen(argv[2], "r");
    }else {
        line = 5;
        fp = fopen(argv[1], "r");
    }
    i = 0;
    while(fgets(buf, SIZE, fp) != NULL){
        i++;
    }
    last = i;
    rewind(fp);
    for(i = 0; i < last - line; i++) {
        fgets(buf, SIZE, fp);
    }
    for(i = last - line; i < last; i++) {
        fgets(buf, SIZE, fp);
        printf("%s", buf);
    }
    fclose(fp);
}

```

```

p16.cpp
#include <stdio.h>
#include <ctype>

main(int argc, char *argv[])
{
    FILE *fp;
    int ccnt, lcnt, wcnt, wflag, c;

    fp = fopen(argv[1], "r");
    ccnt = 0;
    lcnt = 0;
    wcnt = 0;
    wflag = 0;
    while((c = fgetc(fp)) != EOF) {
        ccnt++;
        if(isspace(c)) {
            wflag = 0;
            if(c == '\n')
                lcnt++;
        }
        else if(wflag == 0){
            wflag = 1;
            wcnt++;
        }
    }
    fclose(fp);
    printf("ファイル名 : %s\n", argv[1]);
    printf("文字数 : %d\n行数 : %d\n単語数 : %d\n", ccnt, lcnt, wcnt);
}

```



```

p17.cpp
#include <stdio.h>
#include <ctype>
#define ALPHA 26

main(int argc, char *argv[])
{
    FILE *fp;
    char freq[ALPHA];
    int flag, c, i;

    fp = fopen(argv[1], "r");
    for(i = 0; i < ALPHA; i++)
        freq[i] = 0;
    while((c = fgetc(fp)) != EOF){
        if(isalpha(c))
            freq[tolower(c) - 'a']++;
    }
    for(i = 0; i < ALPHA; i++)
        printf("%c : %5d\n", i + 'a', freq[i]);
    fclose(fp);
}

```

```

p18.cpp
#include <stdio.h>
#define SIZE 100

main(int argc, char *argv[])
{
    FILE *fp1, *fp2, *fp3;
    char buf[SIZE + 1];

    fp1 = fopen(argv[1], "r");
    fp2 = fopen(argv[2], "r");
    fp3 = fopen(argv[3], "a");

    while(fgets(buf, SIZE, fp1) != NULL)
        fputs(buf, fp3);
    while(fgets(buf, SIZE, fp2) != NULL)
        fputs(buf, fp3);

    fclose(fp1);
    fclose(fp2);
    fclose(fp3);
}

```

p19.cpp

```
#include <stdio.h>
#include <string.h>

#define SIZE 100

main(int argc, char *argv[])
{
    FILE *fp;
    char buf[SIZE + 1];
    int lcnt;

    fp = fopen(argv[2], "r");
    lcnt = 0;
    while(fgets(buf, SIZE, fp)){
        lcnt++;
        if(strstr(buf, argv[1]))
            printf("%6d : %s", lcnt, buf);
    }
    fclose(fp);
}
```

p20.cpp

```
#include <stdio.h>
#include <ctype.h>

#define SIZE 16
#define LINE 16

void disp(char *s, int ccnt);

main(int argc, char *argv[])
{
    FILE *fp;
    char s[SIZE];
    int c, ccnt, lcnt;

    fp = fopen(argv[1], "rb");
    ccnt = 0;
    lcnt = 0;
    while((c = fgetc(fp)) != EOF) {
        s[ccnt++] = c;
        if(ccnt == SIZE) {
            disp(s, ccnt);
            if(++lcnt == LINE) {
                putchar('\n');
                getchar();
                lcnt = 0;
            }
            ccnt = 0;
        }
    }
    if(ccnt)
        disp(s, ccnt);
    fclose(fp);
}

void disp(char *s, int ccnt)
{
    static int count = 0;
    int i, j;

    printf("%8.8X : ", count);
    count += ccnt;
    for(i = 0; i < ccnt; i++)
        printf("%02X ", s[i]);
    for(j = 0; j < 3*(SIZE - i); j++)
        putchar(' ');
    putchar(' ');
    for(i = 0; i < ccnt; i++)
        putchar(isprint(s[i]) ? s[i] : '.');
    putchar('\n');
}
```