

p1.cpp

```
#include <stdio.h>

main()
{
    int m, n;

    printf("二つの整数を入力してください ");
    scanf("%d %d", &m, &n);

    while(m != n) {
        if(m > n)
            m = m - n;
        else
            n = n - m;
    }
    printf("最大公約数は%d¥n", m);
}
```

p2.cpp

```
#include <stdio.h>

main()
{
    int m, n, k;

    printf("二つの整数を入力してください ");
    scanf("%d %d", &m, &n);

    do {
        k = m % n;
        m = n;
        n = k;
    }while(k != 0);
    printf("最大公約数は%d¥n", m);
}
```

```
p3.cpp
#include <stdio.h>

main()
{
    char str[20];

    int num, sign = 0, i = 0, j = 0;
    char tmp, data[] = "0123456789";

    printf("整数を入力してください ");
    scanf("%d", &num);

    if(num < 0) {
        sign = 1;
        num = -num;
    }

    do {
        str[j++] = data[num % 10];
        num /= 10;
        i++;
        if(i % 3 == 0 && num != 0) {
            str[j++] = ',';
        }
    }while(num != 0);

    if(sign == 1) {
        str[j++] = '-';
    }
    str[j--] = '\0';

    for(i = 0; i < j ; i++, j--) {
        tmp = str[i];
        str[i] = str[j];
        str[j] = tmp;
    }
    printf("%s¥n", str);
}
```

p4.cpp

```
#include <stdio.h>
#define    N 5

main()
{
    int a[N] = {1999, 4444, 7777, 2222, 9999},
        b[N] = { 111, 6666, 3333, 8888, 1111},
        c[N];
    int i, cy = 0;

    for(i = N - 1; i >= 0; i--) {
        c[i] = a[i] + b[i] + cy;
        if(c[i] > 10000) {
            c[i] = c[i] - 10000;
            cy = 1;
        }
        else {
            cy = 0;
        }
    }

    for (i = 0; i < N; i++)
        printf("%04d ", c[i]);
    printf("¥n");
}
```

p5.cpp

```
#include <stdio.h>
#define    N 5

main()
{
    int a[N] = {1999, 4444, 7777, 2222, 9999},
        b[N] = { 111, 6666, 3333, 8888, 1111},
        c[N];
    int i, brw = 0;

    for(i = N - 1; i >= 0; i--) {
        c[i] = a[i] - b[i] - brw;
        if(c[i] < 0) {
            c[i] = c[i] + 10000;
            brw = 1;
        }
        else {
            brw = 0;
        }
    }

    for (i = 0; i < N; i++)
        printf("%04d ", c[i]);
    printf("¥n");
}
```

p6.cpp

```
#include <stdio.h>
#define CSZ 8
#define RSZ 6

main()
{
    int type, i, j;
    char data[CSZ][RSZ] = { { '#', '#', '#', '#', '#', '#'},
                            {'#', '#', '#', '#', '#', '#'},
                            {'#', '#', '#', '#', '#', '#'},
                            {'#', '#', '#', '#', '#', '#'},
                            {'#', '#', '#', '#', '#', '#'},
                            {'#', '#', '#', '#', '#', '#'},
                            {'#', '#', '#', '#', '#', '#'},
                            {'#', '#', '#', '#', '#', '#'} };

    printf("1:右 90 度回轉 2:左 90 度回轉 3:上下反轉 4:左右反轉 :");
    scanf("%d", &type);

    switch(type) {
        case 1:
            for(i = 0; i < RSZ; i++) {
                for(j = 0; j < CSZ; j++) {
                    putchar(data[CSZ - j - 1][i]);
                }
                putchar('\n');
            }
            break;
        case 2:
            for(i = 0; i < RSZ; i++) {
                for(j = 0; j < CSZ; j++) {
                    putchar(data[j][RSZ - i - 1]);
                }
                putchar('\n');
            }
            break;
        case 3:
            for(i = 0; i < CSZ; i++) {
                for(j = 0; j < RSZ; j++) {
                    putchar(data[CSZ - i - 1][j]);
                }
                putchar('\n');
            }
            break;
        case 4:
            for(i = 0; i < CSZ; i++) {
                for(j = 0; j < RSZ; j++) {
                    putchar(data[i][RSZ - j - 1]);
                }
                putchar('\n');
            }
            break;
    }
}
```

```
p7.cpp
#include <stdio.h>
#define N 4

main()
{
    double a[N + 1] = { 1, 2, 3, 4, 5 };
    double x, f;
    int i;

    printf("x の値: ");
    scanf("%lf", &x);

    f = a[N];
    for(i = N - 1; i >= 0; i--)
        f = f * x + a[i];
    printf("f(%f) = %f¥n", x, f);
}
```

p8.cpp

```
#include <stdio.h>
#include <math.h>
#define f(x) ((x)*(x)*(x) - (x) + 1)
#define g(x) (3*(x)*(x) - 1)
#define EPS 1e-8
#define LIMIT 50

main()
{
    double x = -2.0, dx;
    int k;

    for(k = 1; k <= LIMIT; k++) {
        dx = x;
        x = x - f(x) / g(x);
        if(fabs(x - dx) < fabs(dx) * EPS) {
            printf("x = %f¥n", x);
            break;
        }
    }
    if (k > LIMIT)
        printf("収束しません¥n");
}
```

p9.cpp

```
#include <stdio.h>
#include <math.h>
#define f(x) ((x)*(x)*(x) - (x) + 1)
#define EPS 1e-8
#define LIMIT 50

main()
{
    double low, high, x;
    int k;

    low = -2.0;
    high = 2.0;
    for(k = 1; k <= LIMIT; k++) {
        x = (low + high) / 2;
        if (f(low) * f(x) < 0)
            high = x;
        else
            low = x;
        if (f(x) == 0 || fabs(high - low) < fabs(low) * EPS) {
            printf("x = %f¥n", x);
            break;
        }
    }
    if(k > LIMIT)
        printf("収束しません¥n");
}
```

```

p10.cpp
#include <stdio.h>

long combi(int, int);

main()
{
    int n, r;

    for(n = 1; n <= 5; n++) {
        for (r = 1; r <= n; r++)
            printf("%dC%d = %ld\t", n, r, combi(n, r));
        printf("\n");
    }
}

long combi(int n, int r)
{
    if(r == 0 || r == n)
        return 1L;
    else
        return combi(n - 1, r) + combi(n - 1, r - 1);
}

```

```

p11.cpp
#include <stdio.h>

long fact(int);

main()
{
    int n;
    for (n = 0; n < 13; n++)
        printf("%2d! = %10ld\n",n, fact(n));
}

long fact(int n)
{
    if (n == 0)
        return 1L;
    else
        return n * fact(n - 1);
}

```

```
p12.cpp
#include <stdio.h>

long fibo(long, long, int);

main()
{
    int n;
    long a1 = 1L, a2 = 2L;
    for(n = 1; n <= 20; n++)
        printf("%3d: %ld\n", n, fibo(a1, a2, n));
}

long fibo(long a1, long a2, int n)
{
    if (n == 1)
        return a1;
    else if(n == 2)
        return a2;
    else
        return fibo(a1, a2, n - 1) + fibo(a1, a2, n - 2);
}
```

```
p13.cpp
#include <stdio.h>
#include <stdlib.h>
#define N 100

main()
{
    int i;

    srand(time(NULL));
    for(i = 0; i < N; i++)
        printf("%d ", rand() % 10);
}
```



```

p14.cpp
#include <stdio.h>
#include <stdlib.h>
#define N 100

main()
{
    int i, n, hist[10] = { 0 };

    srand(time(NULL));

    for(i = 0; i < N; i++)
        hist[rand() % 10]++;

    for(i = 0; i < 10 ; i++)
        printf("hist[%d] : %2d¥n", i, hist[i]);
}

```

```

p15.cpp
#include <stdio.h>

void encrypt(const char *, char *);

main()
{
    char *s1 = "Algorithm with C language.";
    char *s2;
    encrypt(s1, s2);
    printf("%s の暗号は%s¥n", s1, s2);
}

void encrypt(const char *str1, char *str2)
{
    while(*str1 != '¥0') {
        *str2 = *str1 + 3;
        str1++;
        str2++;
    }
    *str2 = '¥0';
}

```

```

p16.cpp
#include <stdio.h>

void decrypt(const char *, char *);

main()
{
    char *s1 = "Dojrulwkp#zlwk#F#odqjxdjh1";
    char *s2;
    decrypt(s1, s2);
    printf("%s を復号すると%s¥n", s1, s2);
}

void decrypt(const char *str1, char *str2)
{
    while(*str1 != '¥0') {
        *str2 = *str1 - 3;
        str1++;
        str2++;
    }
    *str2 = '¥0';
}

```

```

p17.cpp
#include <stdio.h>
#define N 10

main()
{
    int a[] = { 56, 25, 67, 88, 100, 61, 55, 67, 76, 56 };
    int i, j, rank[N];

    for(i = 0; i < N; i++) {
        rank[i] = 1;
        for (j = 0; j < N; j++) {
            if(a[j] > a[i])
                rank[i]++;
        }
    }

    printf(" 得点 順位¥n");
    for(i = 0; i < N; i++)
        printf("%5d %5d¥n",a[i], rank[i]);
}

```

```

p18.cpp
#include <stdio.h>
#include <stdlib.h>

main()
{
    int i, j, flag, data[10];

    srand(time(NULL));

    data[0] = rand() % 10;
    for(i = 1; i < 10; i++){
        do {
            data[i] = rand() % 10;
            flag = 0;
            for(j = 0; j < i; j++) {
                if(data[i] == data[j]) {
                    flag = 1;
                    break;
                }
            }
        } while(flag == 1);
    }

    for(i = 0; i < 10; i++)
        printf("%d ", data[i]);
    printf("\n");
}

```

```

p19.cpp
#include <stdio.h>
#include <stdlib.h>

main()
{
    int i, j, w, data[11];

    srand(time(NULL));

    for(i=0; i < 10; i++)
        data[i] = i;

    for (i = 9; i > 0; i--) {
        j = rand() % 10;
        w = data[i];
        data[i] = data[j];
        data[j] = w;
    }

    for(i = 0; i <10; i++)
        printf("%d ", data[i]);
    printf("\n");
}

```

```

p20.cpp
#include <stdio.h>
#include <stdlib.h>
#define N 1000

main()
{
    double x, y, pai;
    int i, in = 0;

    srand(time(NULL));

    for(i = 1; i <= N; i++) {
        x = (double)rand() / RAND_MAX;
        y = (double)rand() / RAND_MAX;
        if(x * x + y * y <= 1)
            in++;
    }
    pai = 4.0 * in / N;
    printf("πの値 = %f¥n", pai);
}

```

```

p21.cpp
#include <GrWin.h>
#include <math.h>
#define PAI 3.14159

main()
{
    int w = 500, h = 500;
    int n, j;
    double x0, y0, x1, y1, L, theta;

    GWopen(0);
    GWsize(-5, &w, &h);
    GWsize(-3, NULL, NULL);
    GWviewport(0.0, 0.0, (double)w / (double)h, 1.0);
    GWindow(0.0, (double)h - 1.0, (double)w - 1.0, 0.0);
    GWclear(-1);

    L = 100;
    for(n = 3; n <= 12; n++) {
        x0 = 200;
        y0 = 450;
        theta = 360.0 / n;
        for(j = 0; j < n; j++) {
            x1 = x0 + L * cos(PAI * (j * theta) / 180.0);
            y1 = y0 - L * sin(PAI * (j * theta) / 180.0);
            GWline(x0, y0, x1, y1);
            x0 = x1;
            y0 = y1;
        }
    }

    GWleave();
}

```

```

p22.cpp
#include <GrWin.h>
#include <math.h>
#define PAI 3.14159

void KochCurve(int, int, int, int, int);

main()
{
    int w = 500, h = 500;

    GWopen(0);
    GWsize(-5, &w, &h);
    GWsize(-3, NULL, NULL);
    GWviewport(0.0, 0.0, (double)w / (double)h, 1.0);
    GWwindow(0.0, (double)h - 1.0, (double)w - 1.0, 0.0);
    GWclear(-1);

    KochCurve(100, 150, 400, 150, 1);
    KochCurve(100, 250, 400, 250, 2);
    KochCurve(100, 350, 400, 350, 3);
    KochCurve(100, 450, 400, 450, 4);

    GWleave();
}

void KochCurve(int x0, int y0, int x1, int y1, int dim)
{
    int x2, y2, x3, y3, x4, y4;

    if(dim <= 0)
        GWline(x0, y0, x1, y1);
    else {
        x2 = (2 * x0 + x1) / 3;
        y2 = (2 * y0 + y1) / 3;
        x4 = (x0 + 2 * x1) / 3;
        y4 = (y0 + 2 * y1) / 3;
        x3 = x2 + (x4 - x2) * cos(PAI / 3) + (y4 - y2) * sin(PAI / 3);
        y3 = y2 - (x4 - x2) * sin(PAI / 3) + (y4 - y2) * cos(PAI / 3);
        KochCurve(x0, y0, x2, y2, dim - 1);
        KochCurve(x2, y2, x3, y3, dim - 1);
        KochCurve(x3, y3, x4, y4, dim - 1);
        KochCurve(x4, y4, x1, y1, dim - 1);
    }
}

```

```

p23.cpp
#include <stdio.h>
#define N 10

main()
{
    int data[] = { 80, 41, 35, 90, 40, 20, 35, 79, 11, 63 };
    int i, j, k, min, s, w;

    for(k = 0; k < N; k++)
        printf("%d ", data[k]);
    printf("¥n");

    for(i = 0; i < N - 1; i++) {
        min = data[i];
        s = i;
        for(j = i + 1; j < N; j++) {
            if(data[j] < min) {
                min = data[j];
                s = j;
            }
        }
        w = data[i];
        data[i] = data[s];
        data[s] = w;

        for(k = 0; k < N; k++)
            printf("%d ", data[k]);
        printf("¥n");
    }
}

```

```

p24.cpp
#include <stdio.h>
#define N 10

main()
{
    int data[] = { 80, 41, 35, 90, 40, 20, 35, 79, 11, 63 };
    int i, j, k, w;

    for(k = 0; k < N; k++)
        printf("%d ", data[k]);
    printf("¥n");

    for(i = 0; i < N - 1; i++) {
        for(j = N - 1; j > i; j--) {
            if(data[j] < data[j - 1]) {
                w = data[j];
                data[j] = data[j - 1];
                data[j - 1] = w;
            }
        }
        for(k = 0; k < N; k++)
            printf("%d ", data[k]);
        printf("¥n");
    }
}

```

```

p25.cpp
#include <stdio.h>
#define N 10

main()
{
    int data[] = { 80, 41, 35, 90, 40, 20, 35, 79, 11, 63 };
    int i, j, k, w;

    for(k = 0; k < N; k++)
        printf("%d ", data[k]);
    printf("¥n");

    for(i = 1; i < N; i++) {
        w = data[i];
        if(data[i - 1] > w) {
            j = i;
            do {
                data[j] = data[j - 1];
                j--;
            } while(j > 0 && data[j - 1] > w);
            data[j] = w;
        }
        for(k = 0; k < N; k++)
            printf("%d ", data[k]);
        printf("¥n");
    }
}

```

```

p26.cpp
#include <stdio.h>
#include <string.h>
#define N 10

main()
{
    struct student {
        char name[20];
        int age;
    };
    struct student data[N] = { { "suzuki", 23 }, { "yamada", 34 }, { "ochiai", 43 },
                                { "nakai", 26 }, { "fukuda", 58 }, { "sasaki", 21 },
                                { "koike", 18 }, { "ueno", 25 }, { "amano", 39 },
                                { "morita", 51 } };
    char key[20];
    int i;

    printf("名前: ");
    scanf("%s", key);

    i = 0;
    while(i < N && strcmp(key, data[i].name) != 0)
        i++;

    if(i < N)
        printf("%s %d¥n", data[i].name, data[i].age);
    else
        printf("見つかりませんでした¥n");
}

```

```
p27.cpp
#include <stdio.h>
#include <string.h>
#define N 10

main()
{
    struct student {
        char name[20];
        int age;
    };
    struct student data[N + 1] = { { "suzuki", 23 }, { "yamada", 34 }, { "ochiai", 43 },
        { "nakai", 26 }, { "fukuda", 58 }, { "sasaki", 21 },
        { "koike", 18 }, { "ueno", 25 }, { "amano", 39 },
        { "morita", 51 } };

    char key[20];
    int i;

    printf("名前: ");
    scanf("%s", key);

    strcpy(data[N].name, key);
    i = 0;
    while(strcmp(key, data[i].name) != 0)
        i++;

    if(i < N)
        printf("%s %d¥n", data[i].name, data[i].age);
    else
        printf("見つかりませんでした¥n");
}
```



```

p28.cpp
#include <stdio.h>
#include <string.h>
#define N 10

main()
{
    struct student {
        char name[20];
        int age;
    };
    struct student data[N] = { { "amano", 39 }, { "fukuda", 58 }, { "koike", 18 },
                                { "morita", 51 }, { "nakai", 26 }, { "ochiai", 43 },
                                { "sasaki", 21 }, { "suzuki", 23 }, { "ueno", 25 },
                                { "yamada", 34 } };

    char key[20];
    int low, high, mid, flag = 0;

    printf("名前:");
    scanf("%s", key);

    low = 0;
    high = N - 1;
    while(low <= high) {
        mid = (low + high) / 2;
        if(strcmp(key, data[mid].name) == 0) {
            flag = 1;
            break;
        }
        if(strcmp(key, data[mid].name) > 0)
            low = mid + 1;
        else
            high = mid - 1;
    }

    if(flag == 1)
        printf("%s %d¥n", data[mid].name, data[mid].age);
    else
        printf("見つかりませんでした¥n");
}

```

```

p29.cpp
#include <stdio.h>
#include <string.h>
#define N 10
#define TSZ 100

int hash(char *);

main()
{
    struct student {
        char name[20];
        int age;
    };
    struct student data[N] = { { "amano", 39 }, { "fukuda", 58 }, { "koike", 18 },
                               { "morita", 51 }, { "nakai", 26 }, { "ochiai", 43 },
                               { "sasaki", 21 }, { "suzuki", 23 }, { "ueno", 25 },
                               { "yamada", 34 } };
    student hdata[TSZ];
    char key[20];
    int i, code;

    for(i = 0; i < TSZ; i++) {
        strcpy(hdata[i].name, "データがありません");
        hdata[i].age = 0;
    }
    for(i = 0; i < N; i++)
        hdata[hash(data[i].name)] = data[i];

    printf("名前:");
    scanf("%s", key);

    code = hash(key);
    printf("%s %d\n", hdata[code].name, hdata[code].age);
}

int hash(char *s)
{
    int n;

    n = strlen(s);
    return (s[0] - 'A' + (s[n / 2 - 1] - 'A') * 26 + (s[n - 2] - 'A') * 26 * 26) % TSZ;
}

```

```
p30.cpp
#include <stdio.h>
#define N 10

main()
{
    int a[] = { 2, 4, 5, 7, 8, 10, 15, 20, 30, 60 },
        b[] = { 1, 6, 11, 18, 25, 33, 45, 55, 78, 88 },
        c[2 * N];
    int i, j, p;

    i = j = p = 0;
    while(i < N && j < N) {
        if(a[i] <= b[j])
            c[p++] = a[i++];
        else
            c[p++] = b[j++];
    }
    while(i < N)
        c[p++] = a[i++];
    while(j < N)
        c[p++] = b[j++];

    for(i=0;i < 2 * N; i++)
        printf("%d ",c[i]);
    printf("¥n");
}
```

```
p31.cpp
#include <stdio.h>
#include <string.h>

char *search(char *, char *);

main()
{
    static char text[] = "This is a pen. That is a pencil.";
    char *p, *key;

    printf("検索文字: ");
    scanf("%s", key);

    p = search(text, key);
    while(p != '\0') {
        printf("%s\n", p);
        p = search(p + strlen(key), key);
    }
}

char *search(char *text, char *key)
{
    int m, n;
    char *p;

    m = strlen(text);
    n = strlen(key);
    for(p = text; p <= text + m - n; p++){
        if(strncmp(p, key, n) == 0)
            return p;
    }
    return NULL;
}
```

```

p32.cpp
#include <stdio.h>
#include <string.h>

char *search(char *, char *);
void replace(char *, char *, char *);

main()
{
    static char text[] = "This is a pen. That is a pencil.";
    char *key, *rep;

    printf("検索文字:");
    scanf("%s", key);
    printf("置換文字:");
    scanf("%s", rep);

    replace(text, key, rep);
    printf("%s\n", text);
}

void replace(char *text, char *key, char *rep)
{
    char *p, buf[128];

    p = search(text, key);
    while(p != '\0') {
        *p = '\0';
        strcpy(buf, p + strlen(key));
        strcat(text, rep);
        strcat(text, buf);
        p = search(p + strlen(rep), key);
    }
}

char *search(char *text, char *key)
{
    int m, n;
    char *p;

    m = strlen(text);
    n = strlen(key);
    for(p = text; p <= text + m - n; p++) {
        if(strncmp(p, key, n) == 0)
            return p;
    }
    return NULL;
}

```

```

p33.cpp
#include <stdio.h>
#define N 5

int stack[N];
int sp = 0;

void push(int);
int pop(void);
void disp(void);

main()
{
    int n, data;

    while(1) {
        printf("1:push 2:pop 3:表示 => ");
        scanf("%d", &n);
        switch(n) {
            case 1:
                if(sp == N) {
                    printf("スタックオーバーフロー¥n");
                    break;
                }
                printf("データ:");
                scanf("%d", &data);
                push(data);
                break;
            case 2:
                if(sp == 0) {
                    printf("データがありません¥n");
                    break;
                }
                pop();
                break;
            case 3:
                disp();
                break;
        }
    }
}

void push(int n)
{
    stack[sp++] = n;
}

int pop()
{
    return stack[--sp];
}

void disp()
{
    int i;

    for(i = 0; i < sp; i++)
        printf("%d ", stack[i]);
    printf("¥n");
}

```

```

p34.cpp
#include <stdio.h>
#define N 10

char stack[N];
int sp = 0;

void push(int);
int pop(void);

main()
{
    char eq[20];
    int i, a, b;

    printf("四則演算式: ");
    scanf("%s", eq);

    for (i = 0; eq[i] != '\0'; i++) {
        switch (eq[i]) {
            case '+':
                a = pop();
                b = pop();
                push(b + a);
                break;

            case '-':
                a = pop();
                b = pop();
                push(b - a);
                break;

            case '*':
                a = pop();
                b = pop();
                push(b * a);
                break;

            case '/':
                a = pop();
                b = pop();
                if(a == 0)
                    break;
                push(b / a);
                break;

            default:
                push(eq[i] - '0');
        }
    }
    printf("%d¥n", pop());
}

void push(int n)
{
    stack[sp++] = n;
}

int pop()
{
    return stack[--sp];
}

```

```

p35.cpp
#include <stdio.h>
#define N 5

int q[N];
int tail = 0;

void enq(int);
int deq(void);
void disp(void);

main()
{
    int n, data;

    while(1) {
        printf("1:enqueue 2:dequeue 3:表示 => ");
        scanf("%d", &n);
        switch(n) {
            case 1:
                if(tail == N) {
                    printf("キューオーバーフロー¥n");
                    break;
                }
                printf("データ:");
                scanf("%d", &data);
                enq(data);
                break;
            case 2:
                if(tail == 0) {
                    printf("データがありません¥n");
                    break;
                }
                deq();
                break;
            case 3:
                disp();
                break;
        }
    }
}

void enq(int n)
{
    q[tail++] = n;
}

int deq()
{
    int w, i;

    w = q[0];
    tail--;
    for(i = 0; i < tail; i++)
        q[i] = q[i + 1];
    return w;
}

void disp()
{
    int i;

    for(i = 0; i < tail; i++)
        printf("%d ", q[i]);
    printf("¥n");
}

```



```

p36.cpp
#include <stdio.h>
#define N 5

int q[N];
int head = 0;
int tail = 0;

void enq(int);
int deq(void);
void disp(void);

main()
{
    int n, data;

    while(1) {
        printf("1:enqueue 2:dequeue 3:表示 => ");
        scanf("%d", &n);
        switch(n) {
            case 1:
                if((tail + 1) % (N + 1) == head) {
                    printf("キューオーバーフロー¥n");
                    break;
                }
                printf("データ:");
                scanf("%d", &data);
                enq(data);
                break;
            case 2:
                if(head == tail) {
                    printf("データがありません¥n");
                    break;
                }
                deq();
                break;
            case 3:
                disp();
                break;
        }
    }
}

void enq(int n)
{
    q[tail] = n;
    tail++;
    tail = tail % (N + 1);
}

int deq()
{
    int w;

    w = q[head];
    head++;
    head = head % (N + 1);
    return w;
}

```

```
void disp()
{
    int i;

    i = head;
    while(i != tail) {
        printf("%d ", q[i]);
        i++;
        i = i % (N + 1);
    }
    printf("\n");
}
```

```

p37.cpp
#include <stdio.h>
#include <stdlib.h>

struct record {
    char name[20];
    char age[5];
    struct record *next;
};

struct record *talloc(void);

main()
{
    struct record *head, *p;
    head = NULL;
    p = talloc();
    printf("レコードを入力 (name age 終了は ctrl+z) : ");
    while(scanf("%s %s", p->name, p->age) != EOF){
        p->next = head;
        head = p;
        p = talloc();
        printf("レコードを入力 (name age 終了は ctrl+z) : ");
    }

    p = head;
    while(p != NULL){
        printf("%-20s%s¥n", p->name, p->age);
        p = p->next;
    }
}

struct record *talloc(void)
{
    return (struct record *)malloc(sizeof(struct record));
}

```

```

p38.cpp
#include <stdio.h>
#include <stdlib.h>

struct record {
    char name[20];
    char age[5];
    struct record *next;
};

struct record *talloc(void);

main()
{
    struct record *head, *p, *old;

    head = talloc();
    old = head;
    p = talloc();
    printf("レコードを入力 (name age 終了は ctrl+z) : ");
    while(scanf("%s %s", p->name, p->age) != EOF){
        old->next = p;
        old = p;
        p = talloc();
        printf("レコードを入力 (name age 終了は ctrl+z) : ");
    }

    p = head->next;
    while(p != NULL){
        printf("%-20s%s¥n", p->name, p->age);
        p = p->next;
    }
}

struct record *talloc(void)
{
    return (struct record *)malloc(sizeof(struct record));
}

```

```

p39.cpp
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct record {
    char name[20];
    char age[5];
    struct record *next;
};
struct record *head;
struct record *talloc(void);
void genlist(void);
void displist(void);
void top(char *);

main()
{
    char key[20];

    genlist();
    displist();

    printf("検索データ (name 終了は ctrl+Z) : ");
    while(scanf("%s", key) != EOF) {
        top(key);
        displist();
        printf("検索データ (name 終了は ctrl+Z) : ");
    }
}

void top(char *key)
{
    struct record *p, *old;

    p = head->next;
    while(p != NULL) {
        if(strcmp(key, p->name) == 0) {
            if(p != head->next) {
                old->next = p->next;
                p->next = head->next;
                head->next = p;
            }
            return;
        }
        old = p;
        p = p->next;
    }
    printf("キーデータが見つかりません\n");
}

void genlist(void)
{
    struct record *p, *old;

    head = talloc();
    old = head;
    p = talloc();
    printf("レコードを入力 (name age 終了は ctrl+Z) : ");
    while(scanf("%s %s", p->name, p->age) != EOF){
        old->next = p;
        old = p;
        p = talloc();
        printf("レコードを入力 (name age 終了は ctrl+Z) : ");
    }
}

```

```
void displist(void)
{
    struct record *p;
    p = head->next;
    while(p != NULL){
        printf("%-20s%s¥n", p->name, p->age);
        p = p->next;
    }
}

struct record *talloc(void)
{
    return (struct record *)malloc(sizeof(struct record));
}
```

```

p40.cpp
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct record {
    char name[20];
    char age[5];
    struct record *next;
};
struct record *head;

struct record *talloc(void);
void genlist(void);
void displist(void);
void link(char *);

main()
{
    char key[20];

    genlist();
    displist();

    printf("検索データ (name) : ");
    scanf("%s", key);
    link(key);

    displist();
}

void link(char *key)
{
    struct record *p, *n;

    n = talloc();
    printf("追加データ (name age) : ");
    scanf("%s %s", n->name, n->age);

    p = head;
    while(p != NULL) {
        if(strcmp(key, p->name) == 0) {
            n->next = p->next;
            p->next = n;
            return;
        }
        p = p->next;
    }
    printf("キーデータが見つかりません\n");
}

void genlist(void)
{
    struct record *p;

    head = NULL;
    p = talloc();
    printf("データ入力 (name age 終了は ctrl+Z) : ");
    while(scanf("%s %s", p->name, p->age) != EOF) {
        p->next = head;
        head = p;
        p = talloc();
        printf("データ入力 (name age 終了は ctrl+Z) : ");
    }
}

```

```
void displist(void)
{
    struct record *p;
    p = head;
    while(p != NULL) {
        printf("%-20s%s¥n", p->name, p->age);
        p = p->next;
    }
}

struct record *talloc(void)
{
    return (struct record *)malloc(sizeof(struct record));
}
```



```

p41.cpp
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct record {
    char name[20];
    char age[5];
    struct record *next;
};
struct record *head;

struct record *talloc(void);
void genlist(void);
void displist(void);
void del(char *);

main()
{
    char key[20];

    genlist();
    displist();

    printf("削除するデータ : ");
    scanf("%s", key);
    del(key);

    displist();
}

void del(char *key)
{
    struct record *p;

    p = head;

    while(p->next != NULL) {
        if(strcmp(key, p->next->name) == 0) {
            p->next = p->next->next;
            return;
        }
        p = p->next;
    }
    printf("キーデータが見つかりません\n");
}

void genlist(void)
{
    struct record *p;

    head = NULL;
    p = talloc();
    printf("データ入力 (name age 終了は ctrl+Z) : ");
    while(scanf("%s %s", p->name, p->age) != EOF) {
        p->next = head;
        head = p;
        p = talloc();
        printf("データ入力 (name age 終了は ctrl+Z) : ");
    }
}

```

```
void displist(void)
{
    struct record *p;
    p = head;
    while(p != NULL){
        printf("%-20s%s¥n",p->name, p->age);
        p = p->next;
    }
}

struct record *talloc(void)
{
    return (struct record *)malloc(sizeof(struct record));
}
```

```

p42.cpp
#include <stdio.h>
#include <string.h>
#define N 20

main()
{
    struct tree {
        int left;
        char name[12];
        int right;
    };
    struct tree data[N] = { { 0, "", 0 },
        { 2, "とうきょう", 3 },
        { 4, "しぶや", 5 },
        { 6, "いけぶくろ", 7 },
        { -1, "しんじゅく", 8 },
        { -1, "たばた", -1 },
        { -1, "うえの", -1 },
        { -1, "あきはばら", -1 },
        { -1, "めぐろ", -1 } };

    char key[12];
    int p, flag;

    printf("検索名前: ");
    scanf("%s", key);

    p = 1;
    flag = 0;
    while(p != -1){
        if(strcmp(key, data[p].name) == 0) {
            flag = 1;
            break;
        }
        else if(strcmp(key, data[p].name) < 0)
            p = data[p].left;
        else
            p = data[p].right;
    }
    if(flag == 1)
        printf("%s は見つかりました¥n", key);
    else
        printf("%s は見つかりません¥n", key);
}

```

```

p43.cpp
#include <stdio.h>
#include <string.h>
#define N 20

main()
{
    struct tree {
        int left;
        char name[12];
        int right;
    };
    struct tree data[N] = { { 0, "", 0 },
        { 2, "とうきょう", 3 },
        { 4, "しぶや", 5 },
        { 6, "いけぶくろ", 7 },
        { -1, "しんじゅく", 8 },
        { -1, "たばた", -1 },
        { -1, "うえの", -1 },
        { -1, "あきはばら", -1 },
        { -1, "めぐろ", -1 } };

    int sp = 9;
    char key[12];
    int p, old, i;

    printf("追加名前: ");
    scanf("%s", key);

    p = 1;
    while(p != -1){
        old = p;
        if(strcmp(key, data[p].name) <= 0)
            p = data[p].left;
        else
            p = data[p].right;
    }

    data[sp].left = -1;
    data[sp].right = -1;
    strcpy(data[sp].name, key);
    if(strcmp(key, data[old].name) <= 0)
        data[old].left = sp;
    else
        data[old].right = sp;
    sp++;

    for(i=1; i < sp; i++)
        printf("%4d%12s%4d¥n",data[i].left, data[i].name, data[i].right);
}

```

```

p44.cpp
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct tree {
    struct tree *left;
    char name[12];
    struct tree *right;
};

struct tree *talloc(void);
struct tree *gentree(struct tree *, char *);
void treewalk(struct tree *);

main()
{
    char data[12];
    struct tree *root;

    root = NULL;
    printf("データ入力 (終了は ctrl+z) : ");
    while (scanf("%s", data) != EOF) {
        root = gentree(root, data);
        printf("データ入力 (終了は ctrl+z) : ");
    }
    treewalk(root);
}

void treewalk(struct tree *p)
{
    if(p != NULL) {
        treewalk(p->left);
        printf("%s¥n", p->name);
        treewalk(p->right);
    }
}

struct tree *gentree(struct tree *p, char *w)
{
    if(p == NULL) {
        p = talloc();
        strcpy(p->name, w);
        p->left = NULL;
        p->right = NULL;
    }
    else if(strcmp(w, p->name) < 0)
        p->left = gentree(p->left, w);
    else
        p->right = gentree(p->right, w);
    return p;
}

struct tree *talloc(void)
{
    return (struct tree *)malloc(sizeof(struct tree));
}

```

```

p45.cpp
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct tree {
    struct tree *left;
    char name[12];
    struct tree *right;
};

struct tree *talloc(void);
struct tree *gentree(struct tree *,char *);
void treewalk(struct tree *);

main()
{
    char data[12];
    struct tree *root;

    root = NULL;
    printf("データ入力 (終了は ctrl+z) : ");
    while (scanf("%s", data) != EOF) {
        root = gentree(root, data);
        printf("データ入力 (終了は ctrl+z) : ");
    }
    treewalk(root);
}

void treewalk(struct tree *p)
{
    if(p != NULL) {
        treewalk(p->right);
        printf("%s¥n", p->name);
        treewalk(p->left);
    }
}

struct tree *gentree(struct tree *p, char *w)
{
    if(p == NULL) {
        p = talloc();
        strcpy(p->name, w);
        p->left = NULL;
        p->right = NULL;
    }
    else if(strcmp(w, p->name) < 0)
        p->left = gentree(p->left, w);
    else
        p->right = gentree(p->right, w);
    return p;
}

struct tree *talloc(void)
{
    return (struct tree *)malloc(sizeof(struct tree));
}

```

```
p46.cpp
#include <stdio.h>
#define N 10

main()
{
    int data[] = { 80, 41, 35, 90, 40, 20, 35, 79, 11, 63 };
    int i, j, k, m, w, gap;

    for(k = 0; k < N; k++)
        printf("%d ", data[k]);
    printf("\n");

    gap = N / 2;
    while(gap > 0) {
        for(m = 0; m < gap; m++) {
            for(i = m + gap; i < N; i = i + gap) {
                j = i - gap;
                while(j >= m && data[j] > data[j + gap]) {
                    w = data[j];
                    data[j] = data[j + gap];
                    data[j + gap] = w;
                    j -= gap;
                }
            }
        }
        gap = gap / 2;

        for(k = 0; k < N; k++)
            printf("%d ", data[k]);
        printf("\n");
    }
}
```

```
p47.cpp
#include <stdio.h>
#define N 10

void quick(int *, int, int);

main()
{
    int data[] = { 80, 41, 35, 90, 40, 20, 35, 79, 11, 63 };

    quick(data, 0, N - 1);
}

void quick(int data[], int left, int right)
{
    int i, j, k, s, w;

    for(k = 0; k < N; k++)
        printf("%d ", data[k]);
    printf("¥n");

    i = left;
    j = right;
    s = data[(left + right) / 2];
    while(1) {
        while(data[i] < s)
            i++;
        while(s < data[j])
            j--;
        if(i >= j)
            break;
        w = data[i];
        data[i] = data[j];
        data[j] = w;
        i++;
        j--;
    }

    for(k = 0; k < N; k++)
        printf("%d ", data[k]);
    printf("¥n");

    if(left < i - 1)
        quick(data, left, i - 1);
    if(j + 1 < right)
        quick(data, j + 1, right);
}
```



```

p48.cpp
#include <stdio.h>
#define N 10

void swap(int *, int *);
void shiftdown(int, int, int *);

main()
{
    int data[] = { 0, 80, 41, 35, 90, 40, 20, 35, 79, 11, 63 };
    int i, n, k;

    for(k = 1; k <= N; k++)
        printf("%d ", data[k]);
    printf("¥n");

    n = N;
    for(i = n / 2; i >= 1; i--)
        shiftdown(i, n, data);

    while(n > 1) {
        swap(&data[1], &data[n]);
        n--;
        shiftdown(1, n, data);
    }
}

void shiftdown(int p, int n, int data[])
{
    int s, k;

    s = 2 * p;
    while(s <= n) {
        if(s < n && data[s + 1] > data[s])
            s++;
        if(data[p] >= data[s])
            break;
        swap(&data[p], &data[s]);
        p = s;
        s = 2 * p;
    }
    for(k = 1; k <= N; k++)
        printf("%d ", data[k]);
    printf("¥n");
}

void swap(int *a, int *b)
{
    int w;

    w = *a;
    *a = *b;
    *b = w;
}

```

```

p49.cpp
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct record {
    struct record *left;
    char name[20];
    char age[5];
    struct record *right;
};

struct record *talloc(void);

main()
{
    struct record *head, *tail, *p;

    tail = NULL;
    printf("データ入力 (name age 終了は ctrl+Z) : ");
    p = talloc();
    while(scanf("%s %s", p->name, p->age) != EOF) {
        p->left = tail;
        tail = p;
        printf("データ入力 (name age 終了は ctrl+Z) : ");
        p = talloc();
    }

    p = tail;
    head = NULL;
    while(p != NULL){
        p->right = head;
        head = p;
        p = p->left;
    }

    printf("順方向リスト");
    p = head;
    while(p != NULL) {
        printf("%-20s%s\n", p->name, p->age);
        p = p->right;
    }

    printf("逆方向リスト");
    p = tail;
    while(p != NULL) {
        printf("%-20s%s\n", p->name, p->age);
        p = p->left;
    }
}

struct record *talloc(void)
{
    return (struct record *)malloc(sizeof(struct record));
}

```

```

p50.cpp
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct tree {
    struct tree *left;
    char name[12];
    struct tree *right;
};
struct tree *talloc(void);
struct tree *gentree(struct tree *,char *);
void treewalk(struct tree *);

main()
{
    char data[12];
    struct tree *root;

    root = NULL;
    printf("データ入力:");
    while(scanf("%s",data) != EOF){
        root = gentree(root, data);
        printf("データ入力:");
    }
    treewalk(root);
}

void treewalk(struct tree *p)
{
    struct connect {
        struct tree *node;
        char *parent;
        char direct;
    } q[128], w[128];
    int i, child, n, level;

    child = 1;
    level = 0;
    q[0].node = p;
    q[0].parent = "root";
    q[0].direct=' ';
    do {
        n = 0;
        printf("レベル %d :¥n", level);
        for(i = 0; i < child; i++) {
            printf("%12s->%c:%12s¥n", q[i].parent, q[i].direct, q[i].node->name);
            if(q[i].node->left != NULL) {
                w[n].parent = q[i].node->name;
                w[n].direct = 'l';
                w[n].node = q[i].node->left;
                n++;
            }
            if(q[i].node->right != NULL) {
                w[n].parent = q[i].node->name;
                w[n].direct = 'r';
                w[n].node = q[i].node->right;
                n++;
            }
        }
        child =n;
        for(i = 0; i < child; i++)
            q[i] = w[i];
        level++;
    }while(child != 0);
}

```

```
struct tree *gentree(struct tree *p,char *w)
{
    if(p == NULL) {
        p = talloc();
        strcpy(p->name, w);
        p->left = NULL;
        p->right = NULL;
    }
    else if(strcmp(w, p->name) < 0)
        p->left = gentree(p->left, w);
    else
        p->right = gentree(p->right, w);
    return p;
}

struct tree *talloc(void)
{
    return (struct tree *)malloc(sizeof(struct tree));
}
```