

```
vi kansu1.c
gcc -o kansu1.out kansu1.c
./kansu1.out
```

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

```
int buy(int x, int y);
```

```
main()
```

```
{
```

```
    int a, b, sum;
```

```
    a = 5;
```

```
    b = 10;
```

```
    sum = buy(a,b);
```

```
    printf("sum = %d\n",sum);
```

```
    return 0;
```

```
}
```

```
int buy(int x, int y)
```

```
{
```

```
    int tmp;
```

```
    tmp = x + y;
```

```
    return tmp;
```

```
}
```

```
[~]$ ./kansu1.out
sum = 15
```

```
vi kansu2.c
gcc -o kansu2.out kansu2.c
./kansu2.out
```

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

```
void replace(int x, int y);
```

```
main()
{
    int a, b;

    a = 5;
    b = 10;

    printf("a = %d, b= %d\n",a,b);
    replace(a,b);
    printf("a = %d, b= %d\n",a,b);
    return 0;
}
```

```
void replace(int x, int y)
{
    int tmp;

    tmp = x;
    x = y;
    y = tmp;
}
```

a と b の値が、
入れ換わって
いない。

```
[~]$ ./kansu2.out
a = 5, b = 10
a = 5, b = 10
```

```
cp kansu2.c kansu3.c
vi kansu3.c
gcc -o kansu3.out kansu3.c
./kansu3.out
```

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

```
void replace(int *x, int *y);          (← 変更)
```

```
main()
```

```
{
```

```
    int a, b;
```

```
    a = 5;
```

```
    b = 10;
```

```
    printf("a = %d, b = %d\n",a,b);
```

```
    replace(&a,&b);          (← 変更)
```

```
    printf("a = %d, b = %d\n",a,b);
```

```
    return 0;
```

```
}
```

```
void replace(int *x, int *y)          (← 変更)
```

```
{
```

```
    int tmp;
```

```
    tmp = *x;          (← 変更)
```

```
    *x = *y;          (← 変更)
```

```
    *y = tmp;          (← 変更)
```

```
}
```

a と b の値が、
入れ換わって
いる。

```
[ a = 5, b = 10 ]$ ./kansu3.out
a = 10, b = 5
```

```
vi pointer1.c
gcc -o pointer1.out pointer1.c
./pointer1.out
```

```
#include <stdio.h>
main()
{
    int a, *b;
    a = 5;
    b = &a;

    printf(" a = %d\n", a);
    printf("&a = %p\n", &a);
    printf(" b = %p\n", b);
    printf("*b = %d\n", *b);

    return 0;
}
```

```
[ ]$ ./pointer1.out
a = 5
&a = 0xbf81991c
b = 0xbf81991c
*b = 5
```

&a と b の値は、この例と異なるが、&a と b の値は等しい (&a=b)。

```
cp pointer1.c pointer2.c
vi pointer2.c
gcc -o pointer2.out pointer2.c
./pointer2.out
```

```
#include <stdio.h>
main()
{
    int a, *b;
    a = 5;
    b = &a;

    printf(" a = %d\n", a);
    printf("&a = %p\n", &a);
    printf(" b = %p\n", b);
    printf("*b = %d\n", *b);

    *b = 100;
    printf("a = %d\n", a);

    return 0;
}
```

```
[ ]$ ./pointer2.out
a = 5
&a = 0xbf9e869c
b = 0xbf9e869c
*b = 5
a = 100
```

b = &a としたことにより、*b を 100 にすると、a の値も 100 になる。

(← 追加)

(← 追加)

```
vi sim1.c
gcc -o sim1.out sim1.c
./sim1.out
```

Fortran の例題プログラム
ex9.for の C 言語版

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

```
main()
```

```
{
```

```
    int i;
```

```
    double t, y, u, dy, du, dt, a;
```

```
    t = 0.0;
```

```
    y = 0.0;
```

```
    u = 0.0;
```

```
    dt = 0.25;
```

```
    a = 9.8;
```

```
    printf("time  positon  velocity\n");
```

```
    for(i=1;i<=20;i++){
```

```
        dy = u * dt;
```

```
        y = y + dy;
```

```
        du = a * dt;
```

```
        u = u + du;
```

```
        t = t + dt;
```

```
        printf("%lf %lf %lf\n",t,y,u);
```

```
    }
```

```
    return 0;
```

```
}
```

```
[time positon velocity]$ ./sim1.out
0.250000 0.000000 2.450000
0.500000 0.612500 4.900000
0.750000 1.837500 7.350000
1.000000 3.675000 9.800000
1.250000 6.125000 12.250000
1.500000 9.187500 14.700000
1.750000 12.862500 17.150000
2.000000 17.150000 19.600000
2.250000 22.050000 22.050000
2.500000 27.562500 24.500000
2.750000 33.687500 26.950000
3.000000 40.425000 29.400000
3.250000 47.775000 31.850000
3.500000 55.737500 34.300000
3.750000 64.312500 36.750000
4.000000 73.500000 39.200000
4.250000 83.300000 41.650000
4.500000 93.712500 44.100000
4.750000 104.737500 46.550000
5.000000 116.375000 49.000000
```

ファイルへの書き込み (fopen, fclose, fprintf)

cp sim1.c sim2.c

vi sim2.c

gcc -o sim2.out sim2.c

./sim2.out

cat sim_data (または vi sim_data)

Fortran の例題プログラム

ex10.for の C 言語版

#include <stdio.h>

main()

{

int i;

double t, y, u, dy, du, dt, a;

FILE *fp;

(← 追加)

t = 0.0;

y = 0.0;

u = 0.0;

dt = 0.25;

a = 9.8;

fp = fopen("sim_data", "w");

(← 追加)

fprintf(fp, "time positon velocity\n");

(← 変更)

for(i=1; i<=20; i++){

dy = u * dt;

y = y + dy;

du = a * dt;

u = u + du;

t = t + dt;

fprintf(fp, "%lf %lf %lf\n", t, y, u);

(← 変更)

}

fclose(fp);

(← 追加)

return 0;

}

sim_data にデータを
書き込む。

cat sim_data により、
sim_data を表示する。

```
[ ]$ ./sim2.out
[ ]$ cat sim_data
time positon velocity
0.250000 0.000000 2.450000
0.500000 0.612500 4.900000
0.750000 1.837500 7.350000
1.000000 3.675000 9.800000
1.250000 6.125000 12.250000
1.500000 9.187500 14.700000
1.750000 12.862500 17.150000
2.000000 17.150000 19.600000
2.250000 22.050000 22.050000
2.500000 27.562500 24.500000
2.750000 33.687500 26.950000
3.000000 40.425000 29.400000
3.250000 47.775000 31.850000
3.500000 55.737500 34.300000
3.750000 64.312500 36.750000
4.000000 73.500000 39.200000
4.250000 83.300000 41.650000
4.500000 93.712500 44.100000
4.750000 104.737500 46.550000
5.000000 116.375000 49.000000
```