

Types (1B)

Copyright (c) 2009-2016 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

a1.c

```
#include <stdio.h>

int main(void) {
    unsigned int m, n, p;
    int i, j, k;

    m = -1;
    n = -3;

    printf("unsigned char m, n, p\n");
    printf("m=%u: %u \n", m);
    printf("n=%u: %u \n", n);

    i = -1;
    j = -3;

    printf("signed char i, j, k\n");
    printf("i=%d: %d \n", i);
    printf("j=%d: %d \n", j);

    printf("m + n  =%u: %u \n", m + n );
    printf("m - n  =%u: %u \n", m - n );
    printf("i + j  =%d: %d \n", i + j );
    printf("i - j  =%d: %d \n", i - j );

    printf("m + j  =%u: %u \n", m + j );
    printf("m - j  =%u: %u \n", m - j );
    printf("i + n  =%u: %u \n", i + n );
    printf("i - n  =%u: %u \n", i - n );

    printf("(m > 0) =%d \n", (m > 0));
    printf("(i > 0) =%d \n", (i > 0));
    printf("(m > n) =%d \n", (m > n));
    printf("(i > j) =%d \n", (i > j));
    printf("(m < 256) =%d \n", (m < 256));
    printf("(i < 256) =%d \n", (i < 256));
}
```

a2.c

```
include <stdio.h>

int main(void) {

    FILE *fp1, *fp2, *fp3;
    int i, m;

    fp1 = fopen("output1.txt", "w");
    fp2 = fopen("output2.txt", "w");

    for (i=0; i<16; ++i) {
        fprintf(fp1, "i= %d \n", i*100);
        fprintf(fp2, "i= %d \n", i*111);
    }

    fclose(fp1);
    fclose(fp2);

    fp1 = fopen("output1.txt", "w");
    fp2 = fopen("output2.txt", "a");
    for (i=0; i<16; ++i) {
        fprintf(fp1, "i= %d \n", i*10000);
        fprintf(fp2, "i= %d \n", i*10101);
    }
    fclose(fp1);
    fclose(fp2);

    fp3 = fopen("input.txt", "r");
    for (i=0; i<16; ++i) {
        fscanf(fp3, " %d", &m);
        printf("i=%d m=%d \n", i, m);
    }
    fclose(fp3);

}
```

a3.c

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

int main(void) {
    char        i, j, k;
    double      x, y, z;

    i= -128;
    j= -125;

    printf("i= %d \n", i);
    printf("j= %d \n", j);

    // abort();
    // error(1, 222, "overflow??? \n");

    // printf("k=i+j : %d \n", k=i+j);
    printf("k=i-j : %d \n", k=i-j);

    x = i = 127;
    y = j = 0;

    printf("k=i/j : %d \n", k=i/j);
    printf("z=x/y : %f \n", z=x/y);

}
```

a3.c

```
#include <stdio.h>

int main(void) {
    unsigned char m, n, p;
    char          i, j, k;

    m = -1; // -1 -> 255
    n = -3; // -3 -> 252

    printf("m=-1 : %u \n", m);
    printf("n=-3 : %u \n", n);

    i = -1;
    j = -3;

    printf("i=-1 : %d \n", i);
    printf("j=-3 : %d \n", j);

    p = m + n;
    k = i + j;

    printf("p= m+n : %u \n", p);
    printf("k= i+j : %d \n", k);

    p = m - n;
    k = i - j;

    printf("p= m-n : %u \n", p);
    printf("k= i-j : %d \n", k);

    /*
    printf("m+j : %u \n", m+j);
    printf("m-j : %u \n", m-j);
    printf("i+n : %u \n", i+n);
    printf("i-n : %u \n", i-n);
    */

    printf("p=m+j : %u \n", p=m+j);
    printf("p=m-j : %u \n", p=m-j);
    printf("p=i+n : %u \n", p=i+n);
    printf("p=i-n : %u \n", p=i-n);
}
```

```
printf("k=m+j : %u \n", k=m+j);  
printf("k=m-j : %u \n", k=m-j);  
printf("k=i+n : %u \n", k=i+n);  
printf("k=i-n : %u \n", k=i-n);  
}
```

a3.c

```
#include <stdio.h>

union B8 {
    long int l;
    double x;
} ;

int main(void) {
    int i = 123;
    double x = 123.0;
    union B8 U;

    U.l = 123;
    printf("U.l=%lx %ld \n", U.l, U.l);

    U.x = 123.0;
    printf("U.l=%lx %ld \n", U.l, U.l);
    printf("U.x=%f \n", U.x);
}
```

a3.c

```
#include <stdio.h>

int main(void) {
    unsigned int m1 = 0xffffffff00;
    unsigned int m2 = 0xf0;
    unsigned int m3 ;

    /*
     int m1 = 0xffffffff00;
     int m2 = 0xf0;
     int m3 ;
    */

    printf("%ld \n", (1L << 32) - 1);

    m3 = m1 - m2;

    printf("m1=%10u m2=%10u m3=%10u\n", m1, m2, m3);
    printf("m1=%10d m2=%10d m3=%10d\n", m1, m2, m3);

    if (m1 > 0) printf("positive number\n");
    else      printf("negative number\n");
}
```


a3.c

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

int main(void) {
    char        i, j, k;
    double      x, y, z;

    i= -128;
    j= -125;

    printf("i= %d \n", i);
    printf("j= %d \n", j);

    // abort();
    // error(1, 222, "overflow??? \n");

    // printf("k=i+j : %d \n", k=i+j);
    printf("k=i-j : %d \n", k=i-j);

    x = i = 127;
    y = j = 0;

    printf("k=i/j : %d \n", k=i/j);
    printf("z=x/y : %f \n", z=x/y);

}
```

a3.c

```
#include <stdio.h>

int main(void) {
    unsigned char m, n, p;
    char          i, j, k;

    m = -1; // -1 -> 255
    n = -3; // -3 -> 252

    printf("m=-1 : %u \n", m);
    printf("n=-3 : %u \n", n);

    i = -1;
    j = -3;

    printf("i=-1 : %d \n", i);
    printf("j=-3 : %d \n", j);

    p = m + n;
    k = i + j;

    printf("p= m+n : %u \n", p);
    printf("k= i+j : %d \n", k);

    p = m - n;
    k = i - j;

    printf("p= m-n : %u \n", p);
    printf("k= i-j : %d \n", k);

    /*
    printf("m+j : %u \n", m+j);
    printf("m-j : %u \n", m-j);
    printf("i+n : %u \n", i+n);
    printf("i-n : %u \n", i-n);
    */

    printf("p=m+j : %u \n", p=m+j);
    printf("p=m-j : %u \n", p=m-j);
    printf("p=i+n : %u \n", p=i+n);
    printf("p=i-n : %u \n", p=i-n);
}
```

```
printf("k=m+j : %u \n", k=m+j);  
printf("k=m-j : %u \n", k=m-j);  
printf("k=i+n : %u \n", k=i+n);  
printf("k=i-n : %u \n", k=i-n);  
}
```


