

TypeCast (1B)

Some codes from cprogramex.wordpress.com

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) {
    int    a;
    int    *p;
    unsigned char *q;

    a = 0xFFEEDDCC;
    p = &a;
    q = (unsigned char *) &a;

    printf("&a= %p a= %x \n", &a, a);
    printf("&p= %p p= %p *p= %x\n", &p, p, *p);
    printf("&q= %p q= %p *q= %x\n", &q, q, *q);

    printf("(q+0)= %p *(q+0)= %x\n", (q+0), *(q+0));
    printf("(q+1)= %p *(q+1)= %x\n", (q+1), *(q+1));
    printf("(q+2)= %p *(q+2)= %x\n", (q+2), *(q+2));
    printf("(q+3)= %p *(q+3)= %x\n", (q+3), *(q+3));
}
```

a2.c

```
#include <stdio.h>

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

int main(void) {
    union B8 U;

    printf("sizeof(long int)=%ld \n", sizeof(long int));
    printf("sizeof(long )=%ld \n", sizeof(long ));
    // I'm using the linux os on a 64-bit computer.

    U.x = 123; // or 123. or 123.0

    printf("123.0 = 0x%016lx \n", U.l);

    U.l = 123;

    printf("123.0 = 0x%016lx \n", U.l);
}
```

a3.c

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

int main(void) {
    int n;
    int i;
    int *p;
    int *q;

    printf("Enter the size of array : ");
    scanf("%d", &n);

    p = (int *) malloc(n * sizeof(int));
    q = (int *) calloc(n , sizeof(int));

    printf("p=%p q=%p \n", p, q);
    for (i=0; i<n; ++i) {
        printf("p[%d]=%5d ", i, p[i]);
        printf("q[%d]=%5d\n", i, q[i]);
    }

    for (i=0; i<n; ++i) {
        p[i] = 100*i;
        q[i] = 200*i;
        printf("p[%d]=%5d ", i, p[i]);
        printf("q[%d]=%5d\n", i, q[i]);
    }

    free(p);
    free(q);

    p = (int *) malloc(n * sizeof(int));
    q = (int *) calloc(n , sizeof(int));

    printf("p=%p q=%p \n", p, q);
    for (i=0; i<n; ++i) {
        printf("uninit p[%d]=%5d ", i, p[i]);
        printf("0-init q[%d]=%5d\n", i, q[i]);
    }

    free(p);
    free(q);
}
```



a3.c

```
int main(void) {
    int n;
    int i;
    int *p;
    // int a[n];
    // array size must be fixed before compilation
    // variable size not allowed

    printf("Enter the size of array : ");
    scanf("%d", &n);

    p = (int *) malloc(n * sizeof(int));

    for (i=0; i<n; ++i)
        p[i] = 100*i;

    printf("sizeof(p)=%ld bytes\n", sizeof(p));
    printf("sizeof(int)=%ld bytes\n", sizeof(int));

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

    free(p);
}
```

a3.c

a3.c

a3.c

a3.c

