```c
#include <stdio.h>
#include "person-record.h"
#include "std-macros.h"

/****
 *
 * This is a testing program to illustrate the use of the different versions of
 * person record defined in person-record.h.
 *
 */
main() {
    /*
     * Declare some persron-record variables, both pointer and non-pointer.
     * Note the compile-time initialization of PersonRecordV3.
     */
    PersonRecordV1 prv1;
    PersonRecordV1* prv1p;
    PersonRecordV2 prv2;
    PersonRecordV2* prv2p;
    PersonRecordV3 prv3 =
        {{"Jane", ’\0’, "Doe"}, 123456789,
         {1, "Main St.", "Holtsville", {’N’, ’Y’}, "USA", 00504}, 25};
    PersonRecordV3* prv3p;
    Name n;
    Address a;

    /*
     * Print the sizes of each version of person record.
     */
    printf("Sizes:\n    prv1=%d, prv1p=%d, prv2=%d, prv2p=%d, prv3=%d, prv3p=%d,\
 prv4=%d, prv4p=%d\n    n=%d, fn=%d, a=%d\n\n",
        sizeof(prv1), sizeof(prv1p), sizeof(prv2), sizeof(prv2p),
        sizeof(prv3), sizeof(prv3p), sizeof(n), sizeof(a));

    /*
     * Set values of prv1.  Print it out a couple times, using the pointer and
     * non-pointer versions of the print function.
     */
    prv1.name = "Jane Doe";
    prv1.id = 123457789;
    prv1.address = "1 Main St.";
    prv1.age = 25;
    printf("prv1 =\n");
    printPersonRecordV1(prv1);
    printf("prv1, again =\n");
    printPersonRecordV1p(&prv1);        /* Same output as preceding print */

    /*
     * Allocate and set values of prv1p.  Print it out.
     */
    prv1p = new(PersonRecordV1);
    prv1p->name = "Jane Doe";
    prv1p->id = 123456789;
    prv1p->address = "1 Main St.";
    prv1p->age = 25;
    printf("*prv1p =\n");
    printPersonRecordV1p(prv1p);
    printf("*prv1p, again =\n");
    printPersonRecordV1(*prv1p);        /* Same output as preceding print */

    /*
     * Set values of prv2.  Print it out.
     */
    strcpy(prv2.name, "Jane Doe");
    prv2.id = 123456789;
    strcpy(prv2.address, "1 Main St.");
    prv2.age = 25;
    /* prv2.name = "Jr Doe";            -- Why doesn’t this work? */
    /* prv2.address = prv1.address;     -- Ditto? */
    printf("prv2 =\n");
    printPersonRecordV2(prv2);
    /* printPersonRecordV1((PersonRecordV1) prv2);  -- Why doesn’t this work? */
    printPersonRecordV1p((PersonRecordV1*) &prv2);  // Ditto

    /*
     * Set prv2p to point to prv2, and change the age of both.  Print both out.
     */
    prv2p = &prv2;
    prv2p->age = 26;
    printf("prv2, changed age =\n");
    printPersonRecordV2(prv2);
    printf("*prv2p =\n");
    printPersonRecordV2(*prv2p);

    /*
     * Allocate a prv3p and initialize it with a raw memory copy of prv3.
     * Print both out.
     */
    prv3p = new(PersonRecordV3);
    memcpy((void*) prv3p, (void*) &prv3, sizeof(prv3));
    printf("*prv3 =\n");
    printPersonRecordV3(prv3);   /* What’s up with the zip code output value? */
    printf("*prv3p =\n");
    printPersonRecordV3(*prv3p); /* Ditto? */

    /*
     * Explanations of the following outputs are to be provided by the reader
     * (i.e., CSC 357 student):
     */
    printf("prv1.address == prv1p->address? -- %s\n\n",  /* Explain the */
       prv1.address == prv1p->address ? "yes" : "no");   /*  output of this */

    printf("%d\n", ((int*) prv1p)[1]);     /* Explain the output of this. */
    printf("%c\n", ((char*) &prv2)[26]);   /* And this. */
    printf("%s\n", &((char*) &prv2)[24]);  /* And this. */
    printf("%d\n", ((int*) &prv2)[5]);     /* And this. */
    printf("%d\n", ((int*) &prv2)[13]);    /* And this. */
    printf("%d\n", prv2.age);
    *((int*)(&((char*) &prv2)[52])) = 3;
    printf("%d\n", prv2.age);              /* And this. */
}
```