

```

1 #include <stdio.h>
2 #include "person-record.h"
3 #include "std-macros.h"
4
5 /****
6 *
7 * This is a testing program to illustrate the use of the different versions of
8 * person record defined in person-record.h.
9 *
10 */
11 main() {
12     /*
13     * Declare some person-record variables, both pointer and non-pointer.
14     * Note the compile-time initialization of PersonRecordV3.
15     */
16     PersonRecordV1 prv1;
17     PersonRecordV1* prv1p;
18     PersonRecordV2 prv2;
19     PersonRecordV2* prv2p;
20     PersonRecordV3 prv3 =
21         {"Jane", '\0', "Doe"}, 123456789,
22         {1, "Main St.", "Holtsville", {'N', 'Y'}, "USA", 00504}, 25};
23     PersonRecordV3* prv3p;
24     Name n;
25     Address a;
26
27     /*
28     * Print the sizes of each version of person record.
29     */
30     printf("Sizes:\n   prv1=%d, prv1p=%d, prv2=%d, prv2p=%d, prv3=%d, prv3p=%d,\n
31     prv4=%d, prv4p=%d\n   n=%d, fn=%d, a=%d\n\n",
32         sizeof(prv1), sizeof(prv1p), sizeof(prv2), sizeof(prv2p),
33         sizeof(prv3), sizeof(prv3p), sizeof(n), sizeof(a));
34
35     /*
36     * Set values of prv1. Print it out a couple times, using the pointer and
37     * non-pointer versions of the print function.
38     */
39     prv1.name = "Jane Doe";
40     prv1.id = 123457789;
41     prv1.address = "1 Main St.";
42     prv1.age = 25;
43     printf("prv1 =\n");
44     printPersonRecordV1(prv1);
45     printf("prv1, again =\n");
46     printPersonRecordV1p(&prv1);          /* Same output as preceding print */
47
48     /*
49     * Allocate and set values of prv1p. Print it out.
50     */
51     prv1p = new(PersonRecordV1);
52     prv1p->name = "Jane Doe";
53     prv1p->id = 123456789;
54     prv1p->address = "1 Main St.";
55     prv1p->age = 25;
56     printf("*prv1p =\n");
57
58     printPersonRecordV1p(prv1p);
59     printf("*prv1p, again =\n");
60     printPersonRecordV1(*prv1p);        /* Same output as preceding print */
61
62     /*
63     * Set values of prv2. Print it out.
64     */
65     strcpy(prv2.name, "Jane Doe");
66     prv2.id = 123456789;
67     strcpy(prv2.address, "1 Main St.");
68     prv2.age = 25;
69     /* prv2.name = "Jr Doe";          -- Why doesn't this work? */
70     /* prv2.address = prv1.address;   -- Ditto? */
71     printf("prv2 =\n");
72     printPersonRecordV2(prv2);
73     /* printPersonRecordV1((PersonRecordV1) prv2); -- Why doesn't this work? */
74     printPersonRecordV1p((PersonRecordV1*) &prv2); // Ditto
75
76     /*
77     * Set prv2p to point to prv2, and change the age of both. Print both out.
78     */
79     prv2p = &prv2;
80     prv2p->age = 26;
81     printf("prv2, changed age =\n");
82     printPersonRecordV2(prv2);
83     printf("*prv2p =\n");
84     printPersonRecordV2(*prv2p);
85
86     /*
87     * Allocate a prv3p and initialize it with a raw memory copy of prv3.
88     * Print both out.
89     */
90     prv3p = new(PersonRecordV3);
91     memcpy((void*) prv3p, (void*) &prv3, sizeof(prv3));
92     printf("prv3 =\n");
93     printPersonRecordV3(prv3);          /* What's up with the zip code output value? */
94     printf("prv3p =\n");
95     printPersonRecordV3(*prv3p); /* Ditto? */
96
97     /*
98     * Explanations of the following outputs are to be provided by the reader
99     * (i.e., CSC 357 student):
100    */
101    printf("prv1.address == prv1p->address? -- %s\n", /* Explain the */
102        prv1.address == prv1p->address ? "yes" : "no"); /* output of this */
103
104    printf("%d\n", ((int*) prv1p)[1]); /* Explain the output of this. */
105    printf("%c\n", ((char*) &prv2)[26]); /* And this. */
106    printf("%s\n", &((char*) &prv2)[24]); /* And this. */
107    printf("%d\n", ((int*) &prv2)[5]); /* And this. */
108    printf("%d\n", ((int*) &prv2)[13]); /* And this. */
109    printf("%d\n", prv2.age);
110    *(int*)(&((char*) &prv2)[52]) = 3;
111    printf("%d\n", prv2.age); /* And this. */

```