```
    {"Lisbon",         3364}, {"London",        3458},
    {"Los Angeles",    2451}, {"Manila",        8498},
    {"Mexico City",    2094}, {"Montreal",       320},
    {"Moscow",         4665}, {"Paris",         3624},
    {"Rio de Janeiro", 4817}, {"Rome",          4281},
    {"San Francisco",  2571}, {"Shanghai",      7371},
    {"Stockholm",      3924}, {"Sydney",        9933},
    {"Tokyo",          6740}, {"Warsaw",        4344},
    {"Washington",      205}};

  printf("Enter city name: ");
  scanf("%80[^\n]", city_name);
  ptr = bsearch(city_name, mileage,
                sizeof(mileage) / sizeof(mileage[0]),
                sizeof(mileage[0]), compare_cities);
  if (ptr != NULL)
    printf("%s is %d miles from New York City.\n",
           city_name, ptr->miles);
  else
    printf("%s wasn't found.\n", city_name);

  return 0;
}

int compare_cities(const void *key_ptr,
                   const void *element_ptr)
{
  return strcmp((char *) key_ptr,
                ((struct city_info *) element_ptr)->city);
}
```

Integer Arithmetic Functions

```
int abs(int j);
long int labs(long int j);
long long int llabs(long long int j);

div_t div(int numer, int denom);
ldiv_t ldiv(long int numer, long int denom);
lldiv_t lldiv(long long int numer,
              long long int denom);
```

abs labs The abs function returns the absolute value of an int value; the labs function returns the absolute value of a long int value.

div The div function divides its first argument by its second, returning a div_t value. div_t is a structure that contains both a quotient member (named quot) and a remainder member (rem). For example, if ans is a div_t variable, we could write

```
ans = div(5, 2);
printf("Quotient: %d Remainder: %d\n", ans.quot, ans.rem);
```