

2.1 While spending the summer as a surveyor's assistant, you decide to write a program that transforms compass headings in degrees (0 to 360) to compass bearings. A compass bearing consists of three items: the direction you face (north or south), an angle between 0 and 90 degrees, and the direction you turn before walking (east or west). For example, to get the bearing for a compass heading of 110.0 degrees, you would first face due south (180 degrees) and then turn 70.0 degrees east ($180.0 - 70.0 = 110.0$). Therefore, the bearing is South 70.0 degrees East. Be sure to check the input for invalid compass headings.

```
#include <stdio.h>

void compass_bearing(double heading) {
    if (heading < 0 || heading >= 360) {
        printf("Invalid compass heading! Please enter a value between
0 and 360.\n");
        return;
    }

    double angle;
    if (heading >= 0 && heading < 90) {
        angle = heading;
        printf("North %.1f degrees East\n", angle);
    } else if (heading >= 90 && heading < 180) {
        angle = 180 - heading;
        printf("South %.1f degrees East\n", angle);
    } else if (heading >= 180 && heading < 270) {
        angle = heading - 180;
        printf("South %.1f degrees West\n", angle);
    } else {
        angle = 360 - heading;
        printf("North %.1f degrees West\n", angle);
    }
}

int main() {
    double heading;
    printf("Enter compass heading (0 to 360): ");
    scanf("%lf", &heading);
    compass_bearing(heading);
    return 0;
}
```

2.2 Write a program to control a bread machine. Allow the user to input the type of bread as W for

White and S for Sweet. Ask the user if the loaf size is double and if the baking is manual. The following table details the time chart for the machine for each bread type. Display a statement for each step. If the loaf size is double, increase the baking time by 50 percent. If baking is manual, stop after the loaf-shaping cycle and instruct the user to remove the dough for manual baking. Use functions to display instructions to the user and to compute the baking time.

Operation	White Bread	Sweet Bread
Primary kneading	15 mins	20 mins
Primary rising	60 mins	60 mins
Secondary kneading	18 mins	33 mins
Secondary rising	20 mins	30 mins
Loaf shaping	2 seconds	2 seconds
Final rising	75 mins	75 mins
Baking	45 mins	35 mins
Cooling	30 mins	30 mins

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

```
void display_steps(const char *bread_type, double factor) {
    printf("Primary kneading: %.1f mins\n", (bread_type[0] == 'W' ? 15
: 20) * factor);
    printf("Primary rising: %.1f mins\n", 60 * factor);
    printf("Secondary kneading: %.1f mins\n", (bread_type[0] == 'W' ?
18 : 33) * factor);
    printf("Secondary rising: %.1f mins\n", (bread_type[0] == 'W' ? 20
: 30) * factor);
    printf("Loaf shaping: 2 seconds\n");
    printf("Final rising: %.1f mins\n", 75 * factor);
}
```

```
void baking_time(const char *bread_type, double factor, int manual) {
    if (manual) {
        printf("Manual baking selected. Please remove the dough after
loaf shaping.\n");
    } else {
        printf("Baking: %.1f mins\n", (bread_type[0] == 'W' ? 45 : 35)
* factor);
        printf("Cooling: 30 mins\n");
    }
}
```

```
int main() {
    char bread_type;
    int is_double, is_manual;
    double factor = 1.0;
```

```

printf("Enter bread type (W for White, S for Sweet): ");
scanf(" %c", &bread_type);
if (bread_type != 'W' && bread_type != 'S') {
    printf("Invalid bread type! Use 'W' or 'S'.\n");
    return 1;
}

printf("Is the loaf size double (1 for Yes, 0 for No): ");
scanf("%d", &is_double);
printf("Is the baking manual (1 for Yes, 0 for No): ");
scanf("%d", &is_manual);

if (is_double) factor = 1.5;

display_steps(&bread_type, factor);
baking_time(&bread_type, factor, is_manual);

return 0;
}

```

2.3 Write a program that interacts with the user like this:

(1) Carbon monoxide

(2) Hydrocarbons

(3) Nitrogen oxides

(4) Nonmethane hydrocarbons

Enter pollutant number: 2

Enter number of grams emitted per mile: 0.35

Enter odometer reading: 40112

Emissions exceed permitted level of 0.31 grams/mile.

Use the table of emissions limits below to determine the appropriate message.

First 50,000 Miles Second 50,000 Miles

carbon monoxide 3.4 grams/mile 4.2 grams/mile

hydrocarbons 0.31 grams/mile 0.39 grams/mile

nitrogen oxides 0.4 grams/mile 0.5 grams/mile

nonmethane hydrocarbons 0.25 grams/mile 0.31 grams/mile

3

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

```

void check_emission(int pollutant, double grams, int odometer) {
    double limit;
    if (pollutant == 1) {
        limit = (odometer <= 50000) ? 3.4 : 4.2;
    } else if (pollutant == 2) {
        limit = (odometer <= 50000) ? 0.31 : 0.39;
    } else if (pollutant == 3) {
        limit = (odometer <= 50000) ? 0.4 : 0.5;
    } else if (pollutant == 4) {
        limit = (odometer <= 50000) ? 0.25 : 0.31;
    } else {

```

```

        printf("Invalid pollutant number.\n");
        return;
    }

    if (grams > limit) {
        printf("Emissions exceed permitted level of %.2f
grams/mile.\n", limit);
    } else {
        printf("Emissions are within the permitted level of %.2f
grams/mile.\n", limit);
    }
}

int main() {
    int pollutant, odometer;
    double grams;

    printf("(1) Carbon monoxide\n(2) Hydrocarbons\n(3) Nitrogen
oxides\n(4) Nonmethane hydrocarbons\n");
    printf("Enter pollutant number: ");
    scanf("%d", &pollutant);
    printf("Enter number of grams emitted per mile: ");
    scanf("%lf", &grams);
    printf("Enter odometer reading: ");
    scanf("%d", &odometer);

    check_emission(pollutant, grams, odometer);
    return 0;
}

```

2.4 Write a program that determines the day number (1 to 366) in a year for a date that is provided as input data. As an example, January 1, 1994, is day 1. December 31, 1993, is day 365. December 31, 1996, is day 366, since 1996 is a leap year. A year is a leap year if it is divisible by four, except that any year divisible by 100 is a leap year only if it is divisible by 400. Your program should accept the month, day, and year as integers. Include a function leap that returns 1 if called with a leap year, 0 otherwise.

```

#include <stdio.h>

int leap(int year) {
    return (year % 4 == 0 && (year % 100 != 0 || year % 400 == 0));
}

int main() {

```

```

    int month, day, year, days_in_month[] = {31, 28, 31, 30, 31, 30,
31, 31, 30, 31, 30, 31};
    int day_number = 0;

    printf("Enter month (1-12): ");
    scanf("%d", &month);
    printf("Enter day (1-31): ");
    scanf("%d", &day);
    printf("Enter year: ");
    scanf("%d", &year);

    if (leap(year)) days_in_month[1] = 29;

    for (int i = 0; i < month - 1; i++) {
        day_number += days_in_month[i];
    }
    day_number += day;

    printf("Day number in the year: %d\n", day_number);
    return 0;
}

```