**computeSalary**

Write a C program that determines the gross pay for each employee in a company. The company pays "straight-time" for the first 160 hours worked by each employee for four weeks and pays "time-and-a-half" for all hours worked in excess of 160 hours. You are given a list of employee Ids (an integer), the number of hours each employee worked for the four weeks, and the hourly rate of each employee. The program should input this information for each employee, then determine and display the employee's gross pay. The sentinel value of –1 is used for the employee *id* to indicate the end of input. Your program should include three functions, apart from the main() function, to handle the input, and the computation of the gross pay. The function prototypes for the functions are given as follows:

```
void readInput(int *id, int *noOfHours, int *payRate);
double computeSalary1(int noOfHours, int payRate);
void computeSalary2(int noOfHours, int payRate, double *grossPay);
```

The function **computeSalary1()** uses call by value for returning the result to the calling function. The function **computeSalary2()** uses call by reference to pass the result through the pointer parameter, grossPay, to the caller.

A sample program template is given below to test the functions:

```
#include <stdio.h>
void readInput(int *id, int *noOfHours, int *payRate);
double computeSalary1(int noOfHours, int payRate);
void computeSalary2(int noOfHours, int payRate, double *grossPay);
int main()
{
  int id = -1, noOfHours, payRate;
  double grossPay;

  readInput(&id, &noOfHours, &payRate);
  while  (id != -1) {
    printf("computeSalary1(): ");
    grossPay = computeSalary1(noOfHours, payRate);
    printf("ID %d grossPay %.2f \n", id, grossPay);
    printf("computeSalary2(): ");
    computeSalary2(noOfHours, payRate, &grossPay);
    printf("ID %d grossPay %.2f \n", id, grossPay);
    readInput(&id, &noOfHours, &payRate);
  }
  return 0;
}
void readInput(int *id, int *noOfHours, int *payRate)
{
  /* Write your code here */
}
double computeSalary1(int noOfHours, int payRate)
{
  /* Write your code here */
```

```
}
void computeSalary2(int noOfHours, int payRate, double *grossPay)
{
  /* Write your code here */
}
```

Some sample input and output sessions are given below:

(1) Test Case 1:

```
Enter ID (-1 to end):
11
Enter number of hours:
155
Enter hourly pay rate:
8
computeSalary1(): ID 11 grossPay 1240.00
computeSalary2(): ID 11 grossPay 1240.00
Enter ID (-1 to end):
12
Enter number of hours:
165
Enter hourly pay rate:
8
computeSalary1(): ID 12 grossPay 1340.00
computeSalary2(): ID 12 grossPay 1340.00
Enter ID (-1 to end):
-1
```

(2) Test Case 2:

```
Enter ID (-1 to end):
11
Enter number of hours:
155
Enter hourly pay rate:
8
computeSalary1(): ID 11 grossPay 1240.00
computeSalary2(): ID 11 grossPay 1240.00
Enter ID (-1 to end):
12
Enter number of hours:
160
Enter hourly pay rate:
8
computeSalary1(): ID 12 grossPay 1280.00
computeSalary2(): ID 12 grossPay 1280.00
Enter ID (-1 to end):
13
Enter number of hours:
200
Enter hourly pay rate:
```

8
computeSalary1(): ID 13 grossPay 1760.00
computeSalary2(): ID 13 grossPay 1760.00
Enter ID (-1 to end):
-1

(3) Test Case 3:
Enter ID (-1 to end):
11
Enter number of hours:
165
Enter hourly pay rate:
8
computeSalary1(): ID 11 grossPay 1340.00
computeSalary2(): ID 11 grossPay 1340.00
Enter ID (-1 to end):
-1