

**CS1010E Programming Methodology**  
Semester 1 2016/2017

Week of 3 – 7 October 2016  
Tutorial 6 Suggested Answers  
**Functions as Procedures**

1. The basis representation theorem states the following:

Every  $n \in \mathbb{Z}^+$  can be uniquely expressed as a sum of terms  $\langle a_i \rangle$  such that

$$n = \sum_{i=0}^k a_i b^i, a_k \neq 0$$

where  $b$  is the base of the number.

For example,  $(130)_{10}$  in base 10 can be expressed as  $(1010)_5$  in base 5 since

$$1 \times 10^2 + 3 \times 10^1 + 0 \times 10^0 = 1 \times 5^3 + 0 \times 5^2 + 1 \times 5^1 + 0 \times 5^0$$

- (a) Define a function `print10toB` that takes in a number  $n$  in the decimal (base-10) system and prints the equivalent number in base  $b$ ,  $2 \leq b \leq 10$ .

```
void print10toB(int n, int b);
```

- (b) Define a function `printB1toB2` that takes in a number  $n$  in base  $b_1$  and prints the equivalent number in base  $b_2$  with  $2 \leq b_1, b_2 \leq 10$ .

```
void printB1toB2(int n, int b1, int b2);
```

```
#include <stdio.h>

void print10toB(int n, int b);
void printB1toB2(int n, int b1, int b2);

int main(void) {
    int n, b1, b2;

    scanf("%d %d", &n, &b1);
    printf("Converting (%d)10 converted to base %d... ", n, b1);
    print10toB(n,b1);

    scanf("%d %d %d", &n, &b1, &b2);
    printf("Converting (%d)%d to base %d... ", n, b1, b2);
    printB1toB2(n,b1,b2);

    return 0;
}
```

```

/*
    Function print10toB a base-10 number n in base-b representation.
*/
void print10toB(int n, int b) {
    int factor = 1;

    while (n/factor > 0) {
        factor = factor * b;
    }

    factor = factor / b;

    while (factor > 0) {
        printf("%d", n/factor);
        n = n % factor;
        factor = factor / b;
    }

    printf("\n");
    return;
}

void printB1toB2(int n, int b1, int b2) {
    int dec = 0, base1 = 1;

    while (n > 0) {
        dec = dec + base1 * (n%10);
        n = n/10;
        base1 = base1 * b1;
    }

    print10toB(dec,b2);
    return;
}

```

2. Write a program that re-arranges three input integers in ascending order. The following main function is given to you.

```

#include <stdio.h>

int main(void) {
    int a, b, c;

    printf("Enter three numbers: ");
    scanf("%d%d%d", &a, &b, &c);
    printf("The numbers reordered: %d %d %d\n", a, b, c);
    return 0;
}

```

- (a) Define a **swap** function that takes in two variables via function output parameters and performs the swap on them.
- (b) In the **main** function, call the **swap** procedure defined in 2a to reorder the three input in ascending order. You need not declare any other variables in **main**.

```
#include <stdio.h>

void swap(int *x, int *y);

int main(void) {
    int a, b, c;

    printf("Enter three numbers: ");
    scanf("%d%d%d", &a, &b, &c);

    if (a > b) {
        swap(&a, &b);
    }
    if (b > c) {
        swap(&b, &c);
    }
    /* c will have the largest value at this point */
    if (a > b) {
        swap(&a, &b);
    }

    printf("The numbers reordered: %d %d %d\n", a, b, c);
    return 0;
}

/*
    swap function exchanges the values referred to by
    function output parameters x and y.

    Precondition:  none
    Postcondition: the two argument variables in the caller
                   have their values swapped.
*/
void swap(int *x, int *y) {
    int temp;

    temp = *x;
    *x = *y;
    *y = temp;
    return;
}
```

3. Given the month and the year, you are required to print the calendar for that month. As an example, the calendar for October 2016 is as follows:

S	M	Tu	W	Th	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

As the solution to this problem is very complex, we will break down the problem into the following parts. You are advised to implement and test each part (by defining a suitable `main` function) before moving on to the next part.

- (a) Define a `isLeapYear` function to determine if a given year is leap.

```
int isLeapYear(int year);
```

- (b) Write a function `daysInMonth` to determine the number of days of a given month (expressed as an integer from 1 to 12) and year.

```
int daysInMonth(int month, int year);
```

Note that we will need to invoke the `isLeapYear` function from within the `daysInMonth` function in order to determine if a year is leap, and thus decide if February of that year has 28 or 29 days.

- (c) Write a function `numDays` to compute the number of days elapsed on a given date since 1 January 1900.

```
int numDays(int day, int month, int year);
```

For example, `numDays(10,1,1900)` is 10, i.e. there are ten days between 1 January 1900 and 10 January 1900, both inclusive. As another example, `numDays(5,10,2012)` is 41186. Use the functions defined in question 3b.

- (d) Write a function `dayOfWeek` to return the day of the week given the date.

```
int dayOfWeek(int day, int month, int year);
```

The days are represented by integer values ranging from 0 for Sunday, to 6 for Saturday. For example, `dayOfWeek(10,1,1900)` returns 3 (i.e. Wednesday), and `dayOfWeek(18,2,1900)` returns 0 (i.e. Sunday). Note that 1 January 1900 falls on a Monday.

- (e) Write a function `printMonth` to print the specific month of a given year.

```
void printMonth(int month, int year);
```

For example, the output for January 1900 is

```
S  M Tu  W Th  F  S
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
```

and the output for October 2016 is

```
S  M Tu  W Th  F  S
                1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
```

Use the conversion specifier `"%3d"` to print an integer justified right within 3 spaces.

- (f) Write a `main` function that repeatedly reads the month and year as input and displays the corresponding calendar. The program stops when a single zero is read as input. A sample run of the program is given in the following page. User input is underlined.

Enter the month and year: 10 2016

```
S  M Tu  W Th  F  S
                1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
```

Enter the month and year: 2 2000

```
S  M Tu  W Th  F  S
        1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29
```

Enter the month and year: 0

```

#include <stdio.h>

int isLeapYear(int year);
int daysInMonth(int month, int year);
int numDays(int day, int month, int year);
int dayOfWeek(int day, int month, int year);
void printMonth(int month, int year);

int main(void) {
    int month, year;

    printf("Enter the month and year: ");
    scanf("%d", &month); // scan the month separately from the year
                        // so as to perform a sentinel check.
    while (month != 0) {
        scanf("%d", &year);
        printMonth(month, year);
        printf("\n");
        printf("Enter the month and year: ");
        scanf("%d", &month);
    }

    return 0;
}

/*
    Function printMonth outputs the monthly calendar for
    a given month and year.
*/
void printMonth(int month, int year) {
    int day = dayOfWeek(1, month, year);
    int numDays = daysInMonth(month, year);
    int i = 0;

    printf(" S  M Tu  W Th  F  S\n");
    for (i = 0; i < day; i++) {
        printf("   ");
    }
    for (i = 1; i < numDays; i++) {
        printf("%3d", i);
        day = (day+1)%7;
        if (day == 0) {
            printf("\n");
        }
    }
    printf("%3d\n", i);
    return;
}

```

```

/*
    Function dayOfWeek returns the day of the week given the
    date. Return value ranges from 0 (Sunday) to 6 (Saturday).
*/
int dayOfWeek(int day, int month, int year) {
    return numDays(day, month, year) % 7;
}

/*
    Function numDays returns the number of days elapsed
    for a given date since 1 January 1900.
*/
int numDays(int day, int month, int year) {
    int total = 0;
    int i;

    for (i = 1900; i < year; i++) {
        if (isLeapYear(i)) {
            total += 366;
        } else {
            total += 365;
        }
    }
    for (i = 1; i < month; i++) {
        total += daysInMonth(i, year);
    }
    total += day;
    return total;
}

/*
    Function daysInMonth returns the number of days in the month
    given the month and year.
*/
int daysInMonth(int month, int year) {
    if (month == 2) {
        if (isLeapYear(year)) {
            return 29;
        } else {
            return 28;
        }
    } else if (month == 4 || month == 6 || month == 9 || month == 11) {
        return 30;
    } else {
        return 31;
    }
}

/*

```

```
    Function isLeapYear returns true
    if year is a leap year; false otherwise.
*/
int isLeapYear(int year) {
    return (year%400 == 0) || ((year%4 == 0 && year%100 != 0));
}
```