

CS11001 Programming and Data Structures, Autumn 2014–2015

Class Test 1

Date: 28–August–2014

Time: 7:00–8:00pm

Maximum Marks: 20

Roll no: _____ Name: _____ Section: _____

[Write your answers in the question paper itself. Be neat and tidy.
Answer all questions. Not all blanks carry equal marks.]

1. Supply single-line answers to the following parts.

(2 × 4)

(a) Let `n`, `i` and `sum` be `int` variables. The user enters a positive value of `n`. Which of the following program segments prints the largest value of `sum`?

- (i) `sum = 0; i = 1; while (++i < n) sum += i; printf("%d", sum);`
- (ii) `sum = 0; i = 1; while (i++ < n) sum += i; printf("%d", sum);`
- (iii) `for (sum = 0, i = 1; i < n; i++) sum += i; printf("%d", sum);`
- (iv) `for (sum = 0, i = 1; i <= n; ++i) sum += i; printf("%d", sum);`

(b) What is printed by the following program?

```
main ()
{
    int x = 0, y = 10, z = 20;
    while (1) {
        x++;
        if (y > z) break;
        y += 4*x; z += 2*x;
    }
    printf("x = %d, y = %d, z = %d", x, y, z);
}
```

(c) What is printed by the following program?

```
main() {
    int x = 1, y = 0, z = 1, t;
    for (t = 0; t < 10; ++t) {
        y += (x) ? z : -z;
        z++; x = !x;
    }
    printf("y = %d", y);
}
```

(d) What is printed by the following program?

```
main ()
{
    int x = 0;
    if (x = 0) printf("Case (a): %d", x);
    else if (x -= 7) printf("Case (b): %d", x);
    else printf("Case (c): %d", x);
}
```

2. A positive integer is called square-free if it is not divisible by the square of any prime number. For example, $98 = 2 \times 7^2$, $99 = 3^2 \times 11$, $100 = 2^2 \times 5^2$ are *not* square-free, whereas 101 (a prime) and $102 = 2 \times 3 \times 17$ are square-free. Your task is to find the divisor m of a positive integer n supplied by the user, such that m is square-free and as large as possible. Indeed, m is the product of all the distinct prime factors of n , each taken only once. For example, for $n = 98, 99, 100, 101, 102$, the values of m will be $14 = 2 \times 7, 33 = 3 \times 11, 10 = 2 \times 5, 101, 102 = 2 \times 3 \times 17$, respectively. Complete the following program to solve this problem. (5)

```
main ()
{
    int n, m, d;

    scanf("%d", &n); /* Assume that a positive integer is entered as n */
    d = 2; m = 1; /* Initialize d (potential divisors of n) and m (the output) */

    while ( _____ ) { /* Supply a condition on n */

        if ( _____ ) { /* if n is divisible by d */
            /* If the above condition is true, then d is prime (see the note below) */

            m = _____; /* Record this prime factor of n in m */
            /* Write a loop to remove all factors of the prime d from n */

            _____
        }
        d++; /* Check the next potential divisor in the next iteration */
    }
    printf("The desired square-free divisor is %d\n", m);
}
```

Notice that in this program, the condition of `if` can be true only when d is prime. This is because if d is composite, then we have taken out all the prime factors of d from n before the division by d is made. For example, we remove all factors of 2 and 3 from n before we divide n by 6, 9, 12, There is no necessity to explicitly check the condition whether d is prime.

3. In this exercise, your task is to evaluate a polynomial $a_0 + a_1x + a_2x^2 + \dots + a_dx^d$ with floating-point coefficients a_i at a floating-point value of x . The user supplies the degree d , the value of x , and then the coefficients a_0, a_1, \dots, a_d . In the following program, the variable `sum` accumulates the desired output value, and the variable `xpower` stores the value of x^i in the i -th iteration. Complete the program. (7)

```
main ()
{
    int i, d;
    float x, a, sum, xpower;

    scanf( _____ ); /* Read both d and x from the user */

    xpower = _____ ; sum = _____ ; /* Initialize */
    /* Loop for reading the coefficients and updating sum and xpower */

    for ( _____ ; _____ ; _____ ) {

        scanf( _____ ); /* Read a_i to a */

        _____ /* Update sum */

        _____ /* Update xpower for next iteration */
    }
    printf("The polynomial evaluates to %f\n", sum);
}
```


