NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING

MID-SEMESTER TEST
AY2015/16 Semester 1

CS1010 PROGRAMMING METHODOLOGY

10 October 2015   Time Allowed: **1 hour 30 minutes**

INSTRUCTIONS

1. This question paper contains **ELEVEN** (11**)** questions and comprises **SIX** (6**)** printed pages, including this page.

2. An **ANSWER SHEET** is provided for you to write the answers. It comprises **TWO (2)** printed pages.

3. Answer **ALL** questions within the space provided on the **Answer Sheet**.

4. Maximum score is **30 marks**.

5. This is an **OPEN BOOK** test.

6. Electronic devices, including but not limited to laptop, electronic dictionary and calculator, are **NOT** allowed.

7. Switch off/silence your mobile phone and keep it out of view.

8. Write your **MATRICULATION NUMBER** on the **Answer Sheet** using **A PEN**.

9. Write your answers legibly with a pen or pencil.

10. Submit only the **Answer Sheet** at the end of the test. You may keep the question paper.

——— **END OF INSTRUCTIONS** ———

SECTION A: Multiple Choice Questions (4 Marks)

Each question has only one correct answer. Write your answers in the boxes provided on the **Answer Sheet**. 1 mark for each correct answer and no penalty for wrong answer.

1. Which of the following variable names are invalid?

    i.   `b_b`
    ii.   `8P`
    iii.   `LOL`
    iv.   `_OvO_`
    v.   `do`
    vi.   `T=T`

    A. (i), (iii) and (v)
    B. (i), (v) and (vi)
    C. (ii), (iii), and (v)
    D. (ii), (iv) and (v)
    E. (ii), (v) and (vi)

2. Which of the following conditions does NOT check whether the values of two **int** variables, **a** and **b**, are the same (assuming that no underflow or overflow occurs)?

    A. `a = b`
    B. `a - b == 0`
    C. `a <= b && a >= b`
    D. `!(a < b || a > b)`
    E. `!(a – b)`

3. What is the final value of **a**?

    ```
    int a = 2, b = 6;
    int *ptr1, *ptr2;

    ptr1 = &b;
    ptr2 = &a;
    *ptr1 += *ptr2;
    ptr1 = ptr2;
    ptr2 = ptr1;
    *ptr1 += *ptr2;
    ```

    A. 10
    B. 8
    C. 6
    D. 4
    E. 12

4. Suppose **arr** is an array of **n** integers. What is the return value of **mystery(arr, n)**?

```
int mystery(int arr[], int n){
    int i;

    for (i = 0; i < n; i++)
        if (arr[i]<arr[i+1]) return 0;

    return 1;
}
```

    A.   It returns 1 if the numbers in **arr** are in ascending order, otherwise 0.
    B.   It returns 1 if the numbers in **arr** are in descending order, otherwise 0.
    C.   It returns 1 if the numbers in **arr** are in strictly ascending order, otherwise 0.
    D.   It returns 1 if the numbers in **arr** are in strictly descending order, otherwise 0.
    E.   None of the above.

SECTION B: Structured Questions (13 Marks)

Write your answers in the boxes provided in the **Answer Sheet**.

5. Spot the syntax errors in the following code fragment.   [3 marks]

```
float mark;
scanf("%f", &mark);

printf("Grade = ");
switch(mark) {
    case 0-49: printf('F');    break;
    case 50-59: printf('D');   break;
    case 60-69: printf('C');   break;
    case 70-79: printf('B');   break;
    case 80-100: printf('A');
}
```

6. What is the final value of **sum**?   [2 marks]

```
int i=4, j, sum=0;

while (i<10){
    j=i;

    while (j<10){
        sum += j;
        if (i>6) break;
        j+=4;
    }
    i+=2;
}
```

7. An if statement is given below which prints text messages based on the values of **day** (an integer in [1, 7]) and **time** (an integer in [0, 23]).

```
if (day <= 5){
    if (time <= 8) printf("Case 1\n");
    else printf("Case 2\n");
} else if (day == 6) {
    if (time <= 8) printf("Case 1\n");
    else printf("Case 4\n");
} else {
    if (time <= 8) printf("Case 3\n");
    else printf("Case 4\n");
}
```

Fill in the missing parts (a)-(c) so that it is equivalent to the given if statement without changing the rest of the code. [3 marks]

```
if (              (a)              )
    printf("Case 1\n");
else if (            (b)            )
    printf("Case 2\n");
else if (            (c)            )
    printf ("Case 3\n");
else printf("Case 4\n");
```

8. After learning about repetition control structure, Adam decided to practice using while-loop by writing a function that takes in a list of 5 integers and outputs the largest integer in the list that is less than 50. For example, given **list** = {76, 35, 98, 49, 23}, the function returns 49. Adam submitted his code to CodeCrunch and found that his code failed the test cases. [3 marks]

```
int largest(int list[]) {
    int i = 1;

    while (list[i] < 50)
        i++;

    return list[i];
}
```

(a) Provide a test case (i.e., an array) that will alert Adam to the incorrect initialization of i to 1.

(b) Provide a test case that will highlight the incorrect logic of the loop condition.

(c) Provide a test case that will highlight the failure to check for array bound.

9. Which **two** code fragments are equivalent for initializing the array **arr**? [2 marks]

```
i.    int arr[2][] = {{1,2,3},{0,0,0}};

ii.   int arr[][3] = {{1,2,3},{0}};

iii.  int arr[2][3];

      for (i = 1; i < 4; i++)
          arr[0][i] = i;

iv.   int arr[2][3] = {{0},{1,2,3}}, i, temp;

      for (i = 0; i < 3; i++){
          temp = arr[0][i];
          arr[0][i] = arr[1][i];
          arr[1][i] = temp;
      }
```

SECTION C: Short Programming Question (13 Marks)
Write your answer in the space provided on the **Answer Sheet**.

10. An **n** x **n** matrix A is *antisymmetic* if $A + A^T$ is a matrix consisting of all 0s, where A^T is the *transpose* of A (*i.e.*, a matrix whose rows are the columns of A).

    For example, $\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ is antisymmetic since $\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$.

    In contrast, $\begin{bmatrix} 0 & 1 \\ -2 & 3 \end{bmatrix}$ is NOT antisymmetic since $\begin{bmatrix} 0 & 1 \\ -2 & 3 \end{bmatrix} + \begin{bmatrix} 0 & -2 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ -1 & 6 \end{bmatrix}$.

    Write a function **int isAntisymmetic(int arr[][10], int n)** which takes in an **n** x **n** 2D array **arr**, where **n** is a positive integer in [1, 10], and returns 1 if **arr** is antisymmetric, otherwise 0. [5 marks]

    (Hint: You do NOT need to compute the transpose of the given array.)

11. In a Toto Game, every participant chooses 6 numbers from 1 to 49. In each draw, 6 winning numbers and 1 additional number are drawn. The prize category for a participant is determined by how many of the chosen numbers match with the winning numbers and the additional number as shown in the table below. (For simplicity's sake, we only consider the top-3 prize categories.)

Category	Numbers matched
1	6 winning numbers
2	5 winning numbers + additional number
3	5 winning numbers

    You may assume that 1) the 6 chosen numbers and 7 drawn numbers have been read in and stored in two **int** arrays called **chosen** and **drawn**, respectively, and 2) the last number in **drawn** is the additional number.

(a) Write an algorithm in pseudo-code or in plain English for **determining the prize category and printing a message accordingly**. [4 marks]

(b) Translate your algorithm in (a) into a code fragment in C. [4 marks]

Sample runs:

```
$ Enter your 6 chosen numbers: 15 3 29 2 42 40
$ Enter 7 drawn numbers: 42 2 15 40 3 29 30
Prize 1!
```
Matched winning numbers: 15, 3, 29, 2, 42, 40.

```
$ Enter your 6 chosen numbers: 15 3 29 2 42 40
$ Enter 7 drawn numbers: 3 15 25 40 2 42 29
Prize 2!
```
Matched winning numbers: 15, 3, 2, 42, 40. Matched additional number: 29.

```
$ Enter your 6 chosen numbers: 15 3 29 2 42 40
$ Enter 7 drawn numbers: 25 3 2 29 15 40 30
Prize 3!
```
Matched winning numbers: 15, 3, 29, 2, 40. Matched additional number: Nil.

```
$ Enter your 6 chosen numbers: 15 3 29 2 42 40
$ Enter 7 drawn numbers: 15 42 2 29 14 7 31
No prize.
```
Matched winning numbers: 15, 29, 2, 42. Matched additional number: Nil.

—— **END OF PAPER** ——