

NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING

MID-SEMESTER TEST

AY2013/2014 Semester 1

CS1010 PROGRAMMING METHODOLOGY

12 October 2013

Time Allowed: **1 hour 30 minutes**

---

**INSTRUCTIONS**

1. This question paper contains **TEN (10)** 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. Write legibly with a pen or pencil.
7. Calculators are allowed, but not laptops, PDAs or other computing devices.
8. Submit only the **Answer Sheet** at the end of the test. You may keep the question paper.
9. Write your **MATRICULATION NUMBER** on the **Answer Sheet** using **A PEN**.

———— **END OF INSTRUCTIONS** ————

**SECTION A: 4 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. What is another way of writing the following expression?

```
sum -= --x - y;
```

- A. `x = x - 1; sum = sum - x + y;`
- B. `x = x - 1; sum = sum - x - y;`
- C. `sum = sum - x - y; x = x - 1;`
- D. `sum = sum - x + y; x = x - 1;`
- E. None of the above.

2. What does the function `f()` compute?

```
// Precond: n >= 0
int f(int a, int n) {
    int i;

    for (i = 1; i <= n; i++)
        a = a + a;

    return a;
}
```

- A.  $2an$
- B.  $a^n$
- C.  $2a^n$
- D.  $2^na$
- E.  $a^{2n}$

3. Given the fact the following code fragment correctly calls the function `f()`, which of the following function prototypes is/are possible for `f()`?

```
int a[]={1,2,3}, b = 2, c = 5;
double d = f(a, b, c);
```

- i. `double f(int [], double, double);`
- ii. `void f(int [], int, int);`
- iii. `int f(int *, double, int);`
- iv. `double f(int, double, int *);`

- A. Only (i) is correct.
- B. Only (i) and (ii) are correct.
- C. Only (i) and (iii) are correct.
- D. Only (iii) and (iv) are correct.
- E. None of the above.

4. Which of the following statements is/are true about arrays?
- i. To initialize an array using an initializer, the number of values in the initializer must be the same as the size of the array.
  - ii. The statement `int arr[5] = {1};` declares an integer array of size 5 and initializes all its elements to 1.
  - iii. For a given array `arr`, when its array name `arr` appears in an expression, it refers to the value of its first element.
  - iv. Initializers can only be used in declaration statements.
- A. Only (ii) is correct.
  - B. Only (i) and (ii) are correct.
  - C. Only (ii) and (iv) are correct.
  - D. Only (i), (ii), and (iii) are correct.
  - E. Only (iv) is correct.

### **SECTION B: Structured Questions (16 Marks)**

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

5. What is the output of the following code fragment? [4 marks]

```
int a;
double b;

a = b = 1 / (float)2;

printf("%d %.2f\n", a, b);

if (3 > a < -1 || a++){
    b = 2 * b + a;
} else {
    b = 2 * b - a;
}

printf("%d %.2f\n", a, b);
```

6. What is the output of the following code fragment? [2 marks]

```
int x = 0, a = 500;

do {
    if (x + a > 250)
        continue;

    x++;
} while (a-- > 0);

printf("%d\n", x);
```

7. What is the output of the following code fragment?

[3 marks]

```
int main(void){
    int x = 100, y = 30, z = f(&x, y);
    printf("%d %d %d\n", x, y, z);
    return 0;
}

int f(int *x, int y){
    y = g(y, x);
    return 2 * *x + y;
}

int g(int x, int *y){
    while (*y >= 20){
        x = x * 2;
        *y = *y - 40;
    }
    return x + 2 * *y;
}
```

8. A **positiveFirst()** function is given below which returns 1 if all the positive numbers (if any) appear before all the non-positive numbers (if any) in an array, or 0 otherwise. Parameter **size** is the number of elements in the array **arr**.

For example, it returns 1 for the array **arr1** = {3, 1, -4, 0, -5} because all the positive numbers (i.e., 3 and 1) in this array appear before the non-positive numbers (i.e., -4, 0 and -5). As another example, it returns 0 for the array **arr2** = {3, 1, -4, 0, 5} because the positive number 5 appears after the non-positive numbers (i.e., -4 and 0) in the array. (The value of **size** is 5 for both examples.)

Fill in the missing parts *M1* - *M3* without changing the rest of the given code. [3 marks]

```
int positiveFirst(int arr[], int size){
    int i = 0;

    while (i < size && M1){
        i++;
    }

    while (i < size){
        if (M2) return M3;
        i++;
    }

    return 1;
}
```

9. Write a function **printSquare(int k)** that prints a square of length **k** (**k** > 1). Three examples of squares are as shown below. The printed square must be left-justified. [4 marks]

For k = 2:	For k = 3:	For k = 5:
**	***	*****
**	* *	* *
	***	* *
		*****

### **SECTION C: Short Programming Question (10 Marks)**

Write your answer in the space provided on the **Answer Sheet**.

10. A fast food restaurant has recently started a promotional campaign for its value meals. During this campaign, one soft toy is given out to each customer who purchases a value meal. There are altogether 5 types of soft toys and the type of soft toy to be given out on a particular day is decided randomly. This campaign runs for an indefinite period of time up to a maximum of 100 days.

Attracted by this campaign, a soft toy lover, Kitty, visits the restaurant starting from day 1 and thereafter every other day to get one soft toy, in the hope of collecting all 5 types of soft toys before the campaign ends.

Write a function **int which\_day(int types[], int days)** to decide when Kitty is able to collect all 5 types of soft toys. The parameter **types** is an array which contains integers between 1 and 5 (both inclusive) indicating the type of soft toy to be given out each day. The parameter **days** is a positive integer indicating the total number of days the campaign runs for. This function returns the day on which Kitty is able to collect all 5 types of soft toys, or -1 if she is unable to do so before the campaign ends.

For example, if **types** = {1, 4, 2, 3} and **days** = 4, it means that soft toy 1, 4, 2 and 3 are given out on day 1, 2, 3 and 4 respectively and the campaign ends after day 4. In this case, Kitty is only able to collect toys 1 and 2. Therefore the function should return -1.

As another example, if **types** = {1, 1, 2, 2, 3, 3, 4, 4, 5, 5} and **days** = 10, Kitty will be able to collect all 5 types of toys on day 9. Therefore, the function should return 9.

- Write an algorithm in pseudo-code or plain English for this function. [5 marks]
- Write the code for this function based on the algorithm in (a). The rest of the program is given on the next page and should not be changed. [5 marks]

```
#include <stdio.h>
#define TOY_TYPE 5
#define MAX_DAY 100

int which_day(int [], int);

int main(void){
    int day, type, count=0;
    int types[MAX_DAY] = {0};

    scanf("%d", &type);

    while (type != -1) {
        types[count] = type;
        count++;
        scanf("%d", &type);
    }

    day = which_day(types, count);

    if (day == -1){
        printf("Kitty is unable to collect all types of
toys.\n");
    } else {
        printf("Kitty is able to collect all types of toys on
day %d.\n", day);
    }

    return 0;
}

int which_day(int types[], int days){
    // Complete this function
}
```

———— **END OF PAPER** ————