

NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING

MID-SEMESTER TEST

AY2016/17 Semester 1

CS1010 PROGRAMMING METHODOLOGY

8 October 2016

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. 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 (5 Marks)

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

1. Which of the following variable names are valid?

- i. 10_min_block
- ii. g.s.t.
- iii. Float
- iv. _score
- v. print

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

2. Which of the following statements is correct?

- A. If a condition is evaluated to a negative value, then the condition is false.
- B. Given two integers **a** and **b**, for the operation **a%b** to work, **b** must be non-zero.
- C. **true** is a reserved keyword in ANSI C.
- D. The statement **int arr[5] = {1};** initializes all 5 elements in **arr** to 1.
- E. All of the above are incorrect.

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

```
float i = 1.5;

switch ((int)i) {
    case 1:
        printf("%.1f\n", i); break;
    case 2:
        printf("%.1f\n", i); break;
    case 3:
        printf("%.1f\n", i);
}
```

- A. There is a compilation error.
- B. The value 1.0 is printed.
- C. The value 1.5 is printed.
- D. The value 2.0 is printed.
- E. None of the above options is correct.

4. Which of the following statements generates a random integer in the range [-200, 200]?

- A. `rand() - 200`
- B. `rand() % 200`
- C. `rand() % 400 - 200`
- D. `rand() % 400 - 201`
- E. `rand() % 401 - 200`

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

```
int i = 3, j;
for ( ; i >= 0; i--)
    for (j = 0; j < i; j++)
        printf("*");
printf("\n");
```

- A. `*****`
- B. `**`
`**`
`**`
- C. `*`
`**`
`***`
- D. `***`
`**`
`*`
- E. It is a compilation error to omit the initialization of `i` in the `for` statement.

SECTION B: Structured Questions (8 Marks)

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

6. Assuming **x**, **y** and **z** are integer variables, rewrite the following code fragment without using loops, continue and break statements. [3 marks]

```
do {
    if (!x) continue;
    else y++;

    z = x + y;

    break;
} while(x);
```

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

```
int arr[5] = {1, 2, 3};
int *p1, *p2;

p1 = arr;
p2 = &arr[3];

printf("%d\n", (*p1)++ + --(*p2));
```

8. Complete the following function **arrangeArray** which 1) takes in an array **arr** and its **size**, and 2) places the minimum value to the right of the maximum value in **arr** by swapping the minimum value with the element to the right of the maximum value.

The swap is performed in a wrap-around manner: if the maximum value happens to be the last element, the minimum will be placed at the beginning of the array.

Two examples of the swap are as shown below with the minimum and maximum values highlighted in **bold**:

arr	arr after invoking arrangeArray
{1, 2, 5 , 3, 0 }	{1, 2, 5 , 0 , 3}
{3, 1 , 7, 6, 9 }	{ 1 , 3, 7, 6, 9 }

You may assume that the values in **arr** are distinct. The code for finding the minimum and maximum values in **arr** is given. You are NOT allowed to modify the existing code or create additional variables/arrays. [3 marks]

```
void arrangeArray(int arr[], int size) {
    int i, min_pos = 0, max_pos = 0, temp;

    for (i = 1; i < size; i++) {
        if (arr[i] < arr[min_pos]) min_pos = i;
        if (arr[i] > arr[max_pos]) max_pos = i;
    }

    // Write your code for swapping

}
```

SECTION C: Short Programming Question (17 Marks)

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

9. Write a function `void rotateMatrix(int mtx[][10], int size)` which 1) takes in a square matrix `mtx` and its `size`, and 2) rotates the elements in `mtx` counter-clockwise by 90° .

For example, as shown below on the left, `mtx` is a 3 x 3 square matrix (*i.e.*, `size` is 3). After rotating `mtx` counter-clockwise by 90° , it becomes the matrix on the right.

$$mtx = \begin{bmatrix} 1 & 4 & 3 \\ 5 & -1 & 9 \\ -2 & 7 & 0 \end{bmatrix} \quad \text{rotated } mtx = \begin{bmatrix} 3 & 9 & 0 \\ 4 & -1 & 7 \\ 1 & 5 & -2 \end{bmatrix}$$

You may assume that the maximum size of the matrix is 10. You are NOT allowed to create additional arrays. [6 marks]

(Hint 1: One possible solution involves swapping a group of related elements each time. All elements in the group should be at the right place after the swap.)

(Hint 2: Another possible solution involves flipping the whole matrix more than once. Note that a matrix can be flipped horizontally, vertically and diagonally).

10. **PI on the fly**: One way to calculate the constant pi (π) is given by the formulae as shown below, where each fraction (without the sign) is considered a term.

$$\pi = \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \dots$$

Write a program that repeatedly reads in an integer `noterms`, which represents the number of terms to be used for calculation. As long as `noterms` is positive, the program calculates and prints the value of π . For example, if `noterms` is 3, the value of π is calculated as:

$$\pi = \frac{4}{1} - \frac{4}{3} + \frac{4}{5} = 3.466667$$

To avoid re-computing the individual terms every time, these terms should be stored in an array once they are computed. For example, after the above calculation, the first 3 elements in the array should be 4.0, 1.333333 and 0.8, respectively. There should be no need to re-compute these terms in any subsequent calculations.

In your program, include a function `calculatePI` that takes in

- an integer `noterms`, which is the current input,
- an array `arr`, which contains the terms computed before the current input, and
- an integer `maxterm`, which is the maximum value of `noterms` entered before the current input. (Hint: This is also the number of terms computed in `arr`.)

This function performs the task of populating the terms in `arr`, as well as calculating and returning the value of π .

The program loops repeatedly until 0 is entered. You may assume the input to the program is an integer in $[0, 100000]$. *(The question continues on the next page...)*

- a) Write an algorithm in pseudocode for the above program.

You should write the algorithm for **calculatePI** separately and indicate how it is used in the main algorithm. [5 marks]

- b) Translate the algorithm in (a) to a C program. Marks will be deducted if your program does not follow your algorithm closely.

You do NOT need to include preprocessor directives and function prototypes in your answer (*i.e.*, you may start directly with the main function). [6 marks]

A sample run of the program is shown below with user input highlighted **in bold**.

```
Enter number of terms: 1
The value of pi is: 4.000000
Enter number of terms: 100
The value of pi is: 3.131593
Enter number of terms: 5
The value of pi is: 3.339683
Enter number of terms: 100000
The value of pi is: 3.141583
Enter number of terms: 0
Thank you and goodbye!
```

———— **END OF PAPER** ————