

NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING

PLACEMENT EXAMINATION FOR CS1010/CS1010E

12 July 2013

Time allowed: 2 hours

INSTRUCTIONS TO CANDIDATES

1. This examination paper consists of **FIFTEEN (15)** questions and comprises **TWENTY ONE (21)** printed pages.
2. This is a **closed book** examination. Calculators are not allowed.
3. Answer all questions.
4. You may use **2B pencil** to write your programs.
5. Fill in your NUS Student Number, or if you have not obtained your NUS Student Number, your NUS Application Number below:

--	--	--	--	--	--	--	--	--	--

6. Fill in your Name below:

FOR INTERNAL USE ONLY

<i>Qn</i>	<i>Maximum</i>	<i>Mark</i>
1-10	20	
11	6	
12	8	
13	12	
14	26	
15	28	
Total	100	

Important notes:

- Your programs should exhibit signs of good programming practice. Departure from it may result in some penalty to be incurred.
- Use 2B pencil to write your programs to minimise untidy writing.
- You should try to write within the boxes provided, which have sufficient space to contain your code. Marks may be deducted if your code is longer than necessary. The last two blank pages are to be used only if absolutely necessary.

Questions 1 – 10:

Each question is worth 2 marks. For multiple-choice questions, please write your answers in the brackets provided.

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

```
int n = 50, count = 0;
while (n) {
    n /= 2;
    count++;
}
printf("%d\n", count);
```

- (A) 25
 (B) 7
 (C) 6
 (D) No output as there is an infinite loop
 (E) No output as there is a compilation error ()

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

```
int a = 1, b = 2, c = 3, d = 4, e;
e = (a > b) ? ((b > c) ? a : b) : ((a > c) ? c : d);
printf("%d\n", e);
```

- (A) 1
 (B) 2
 (C) 3
 (D) 4
 (E) No output as there is a compilation error ()

3. There is a string function in <string.h>, let's call it *f*, with the following documentation:

```
char *f(char *str2, char *str1);  
Locates the first occurrence of string str1 in str2 and returns a  
pointer to the beginning of the first occurrence.
```

What is the name of the function?

- (A) strstr
- (B) strtok
- (C) strchr
- (D) strcmp
- (E) strspn ()

4. Given the following list of numbers:

3, 5, 1, 6, 4, 2

One of the three sorting algorithms – bubble sort, insertion sort, and selection sort – is used to sort the above list. After 2 passes, the list becomes:

1, 2, 3, 6, 4, 5

What algorithm is used?

- (A) Bubble sort
- (B) Insertion sort
- (C) Selection sort ()

5. Given the following function:

```
int mysterious(int a[], int size) {  
    int i;  
    for (i = 1; i < size; i++)  
        if (a[i] < a[i-1])  
            return 0;  
    return 1;  
}
```

Suppose the function is called with **mysterious(arr, 6)**. Which of the following is the correct content of **arr** (left-most value is arr[0]) such that the function returns 1?

- (A) 4, 7, 2, 3, 1, 6
- (B) 10, 11, 18, 18, 18, 20
- (C) 23, 25, 23, 25, 23, 25
- (D) 39, 37, 35, 33, 32, 31
- (E) None of the above. ()

6. Given the following function:

```
int f(int n) {
    return (n%100)*10 + n/100;
}
```

What value is returned by the following call: `f(f(35) - f(216))`?

7. Given the following function `g` where `size` is the number of elements in the array `arr`, assuming that `size` is at least 1.

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

    for (i = 1; i < size; i++)
        if (arr[i] < abc)
            abc = arr[i];

    return abc;
}
```

Explain in a single line what the function `g` returns.

The variable `abc` is not appropriately named. Suggest a good name for it.

8. Given the following code fragment:

```
#define MAX 10
#define MIN 3
#define N MAX - MIN
printf("%d %d\n", N, 2*N);
```

What is the output?

The second value printed is not twice the first value printed. If you want the second value printed to be twice the first value printed, how would you change one of the `#define` statements?

9. Given the following recursive function, where n is a positive integer.

```
int recurse(int n) {
    if (n == 1)
        return 3;
    else if (n == 2)
        return 7;
    else if (n == 3)
        return 15;
    else
        return 2 * recurse(n-1) + 1;
}
```

Simplify the function so that it computes the same result as above. Your function should remain recursive.

10. The following code fragment is given.

```
typedef struct {
    int x, y; // x- and y-coordinates of a point
} pt_t;

// Area of rectangle with 2 opposite vertices P and Q
int area(pt_t P, pt_t Q) {
    return (Q.x - P.x) * (Q.y - P.y);
}
```

The function **area()** computes the area of a rectangle with a pair of opposite vertices P and Q , assuming that the rectangle has sides that are parallel to either the x - or y -axis. However, the function works only if P and Q are the bottom-left and top-right vertices of the rectangle; it does not work if P and Q are the top-left and bottom-right vertices.

Rewrite only the **return** statement using an appropriate library function such that the function works for any pair of opposite vertices.

11. [6 marks]

Given the following program, write out its output.

```
#include <stdio.h>

int func1(int, int *);
float func2(float *, float);

int main(void) {
    int a = 3, b = 7;
    float x = 1.6, y = 5.9;

    x = func1(a, &b);
    y = func2(&x, y);

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

    return 0;
}

int func1(int x, int *y) {
    x += *y;
    *y += x;
    return x;
}

float func2(float *a, float b) {
    *a *= (int) b;
    b *= (int) *a;
    return *a;
}
```

Write the output in the box below:

a =
b =
x =
y =

Total marks for Q11:

/ 6

12. [8 marks]

Write a function **convert(char *str)** that takes in a string **str** in its parameter and perform the following:

- Convert all lower-case letters in **str** to their upper-case equivalent. You are to use a library function for this.
- Remove all vowels in **str**.

A sample run is shown below:

```
Enter text: On her 5th birthday, Mary received a Barbie doll.
Converted : N HR 5TH BRTHDY, MRY RCVD BRB DLL.
```

The **main()** function is given, which you should not modify. Complete the function **convert(char *str)**. You need to include the appropriate header file(s) in your code.

```
#include <stdio.h>
#include <string.h>
// include other header file(s) below if necessary

#define MAXLEN 80

void convert(char *);

int main(void) {
    char line[MAXLEN+1];
    int len;

    printf("Enter text: ");
    fgets(line, MAXLEN+1, stdin);
    len = strlen(line);
    if (line[len-1] == '\n')
        line[len-1] = '\0';

    convert(line);
    printf("Converted : %s\n", line);

    return 0;
}
```

12. (continued... page 2 of Q12)

```
// complete the convert() function below  
void convert(char *str) {
```

```
}
```

Total marks for Q12: / 8

13. [12 marks]

(Note that no mark will be awarded if your functions are not recursive.)

The function **pow(x, n)** that computes x^n where both x and n are integers, and n non-negative, can be implemented using the following recurrence relation:

$$x^n = \begin{cases} 1 & \text{if } n = 0 \\ x \cdot x^{n-1} & \text{if } n > 0 \end{cases}$$

- (a) Write a recursive function **pow(x, n)** to compute x^n . [4 marks]

- (b) We could improve efficiency if we make use of $x^{n/2}$ in our recursion. Write a recursive function **newPow(x, n)** to compute x^n . [6 marks]

13. *(continued... page 2 of Q13)*

(c) Draw the trace tree for **newPow(4, 3)**.

[2 marks]

14. [26 marks]

You are given an $n \times m$ rectangular plot (n rows and m columns), where n and m are at most 10. On this plot, you are to tile some $p \times q$ rectangular motifs (p rows and q columns), where p and q are also at most 10, from top to bottom, left to right.

Both the plot and the motif are given as 2-dimensional integer arrays, whose elements are single digits 0 through 9.

You are not to use any string or string-related functions in this question.

The following 2 examples show how the tiling should be done.

Example #1: A 2×3 motif on a 5×8 plot.

5	1	3
2	0	7

2×3 motif

5	1	3	5	1	3	5	1
2	0	7	2	0	7	2	0
5	1	3	5	1	3	5	1
2	0	7	2	0	7	2	0
5	1	3	5	1	3	5	1

5×8 plot

Example #2: A 5×3 motif on a 3×4 plot.

8	1	3
6	9	0
2	6	1
4	6	4
0	2	1

5×3 motif

8	1	3	8
6	9	0	6
2	6	1	2

3×4 plot

You are to write a complete C program to read data from a text file called "**motif.in**". Your program then prints the resulting plot.

To help you write your program, we have split the question into parts. Please read through all the parts before you start writing your code.

All functions should have appropriate return type and parameters. You are NOT allowed to use global variables.

- (a) You are given a structure type **dim_t** as shown in page 13, to hold the dimension (number of rows and columns). Using **typedef**, you are to define another structure type called **grid_t**, which is used to declare the motif and plot variables. The structure **grid_t** should contain two components: the dimension and a 2-dimensional integer array to hold the data. [2 marks]

14. (continued... page 2 of Q14)

- (b) A text file called “**motif.in**” contains data about the plot and motif. On the first line are two integers indicating the number of rows and columns for the plot. On the second line are two integers indicating the number of rows and columns for the motif. Subsequent lines contain the data for the motif.

The example below shows the content of “**motif.in**” that corresponds to example #1 above. Note that the content of the file is not fixed; it may be changed to test the program.

```
5 8
2 3
5 1 3
2 0 7
```

Write a function **readFile(...)** to read data from a text file “**motif.in**”. All data in the text file must be read in this function, and after this function is called, the file should not be read again by the program. [6 marks]

- (c) Write a function **fillPlot(...)** to tile the motif in the plot as described above. [10 marks]

- (d) Write a function **printGrid(...)** to print the filled plot. [4 marks]

- (e) Write the **main()** function which performs the following: [4 marks]

- Call the **readFile(...)** function to read data from the text file “**motif.in**”.
- Call the **fillPlot(...)** function to tile the motif in the plot.
- Call the **printGrid(...)** function to print the filled plot.

The output corresponding to example #1 above is shown below.

```
5 1 3 5 1 3 5 1
2 0 7 2 0 7 2 0
5 1 3 5 1 3 5 1
2 0 7 2 0 7 2 0
5 1 3 5 1 3 5 1
```

14. (continued... page 3 of Q14)

Write your program below.

```
#include <stdio.h>
#define MAX 10

typedef struct {
    int rows, cols;
} dim_t;

// Define grid_t below -- part (a)


// Write the readFile(...) function below -- part (b)
```

14. (continued... page 4 of Q14)

// Write the fillPlot(...) function below -- part (c)

14. (continued... page 5 of Q14)

```
// Write the printGrid() function below -- part (d)
```

```
// Write the main() function below -- part (e)
int main(void) {
```

```
    return 0;
}
```

Total marks for Q14: / 26

15. [28 marks]

Write a complete C program to randomly generate 100 distinct non-negative integers that are multiples of 3 and in the range [0, 100000] and load them into a 100-element integer array `arr`. The program then determines and outputs the smallest difference among all pairs of elements in the array.

For example, assuming that we have a 5-element array instead of a 100-element array, and it contains these 5 values: 81, 333, 24, 300, and 513, then the smallest difference among all pairs of elements in the array is 33 (the difference between 333 and 300).

Given that the range of the random numbers to be generated is known, we expect you to use an efficient algorithm to check for uniqueness while generating the required random numbers. Likewise, we expect that you use an efficient algorithm to determine the smallest difference among all pairs of elements in the array. Full credit will be awarded only if your program meets these expectations. [*Hint*: Use an additional array.]

In case you think that your code is not self-explanatory enough, you may explain the algorithms in the box below. Your explanation will not gain any mark, but it may help make your code easier to understand.

Explanation of your code (only if you think it is necessary):

15. *(continued... page 2 of Q15)*

Write your program below. You should write a modular program that contains two functions: one to generate the distinct random numbers, and one to determine the smallest difference among all pairs of random numbers written into the array.

15. *(continued... page 3 of Q15)*
Continue your program below if necessary.

15. *(continued... page 4 of Q15)*

Continue your program below if necessary.

Total marks for Q15: / 28

(This page is used only if you need more space to write your answers.
Please indicate the question number.)

(This page is used only if you need more space to write your answers.
Please indicate the question number.)

~~~ END OF PAPER ~~~