

The American University in Cairo
Computer Science & Engineering Department
CSCE 106 "Fundamentals of Computer Science"

Instructor: Dr. Khalil	Final Exam	Fall 2013
Last Name :		ID:
First Name:		Form I Section No.: ()

EXAMINATION INSTRUCTIONS

- * Do not turn this page until asked to do so.
- * Exam time is **120** minutes.
- * Put the answers on the same question sheet, do not use any additional papers, even for scratch.
- * Write your name, ID, section no. in the indicated places.
- * Read the exam instructions.
- * Read the honesty policy.
- * Sign the following statement.

Academic Integrity Policy

Cheating in Exams is a violation of the honesty policy of AUC. Whispering, talking, looking at someone else's paper, or copying from any source is considered cheating. Any one who does any of these actions or her/his answers indicates that she/he did any of them, will receive a punishment ranging from zero in this exam to failing the course. If repeated, it may lead to dismissal from AUC.

I have read the honesty policy and exam instructions and I am presenting this exam as entirely my effort.

Signature: _____

DO NOT USE THIS SECTION

Question	Points	Grade
1	10	
2	20	
3	20 + 3	
4	15 + 3	
5	15	
6	20	
Total	100	

Question 1 (10 points)

Tick only one possible answer for each of the following:

1) The Boolean expression $!(x > y)$ can be simplified into:

- a. $(x < y)$ ☐
- b. $(x \geq y)$ ☐
- c. True ☐
- d. none of the above ☐

2) Intellectual Property:

- a. A form of protection that can be granted by the government for an invention; gives exclusive rights of an invention to its inventor for 20 years. ☐
- b. Standards of moral conduct as they relate to computer use. ☐
- c. The rights to which creators of original creative works (such as software, artistic, literary works, inventions, and more) are entitled. ☐
- d. The legal right to sell, publish, or distribute an original software, artistic or literary work; is held by the creator of a work as soon as it exists in physical form. ☐

3) The value of $((x - 4) != 4) \&\& ((x - 4) == 4)$ is:

- a. $(x - 4)$ ☐
- b. False ☐
- c. $(x + 4)$ ☐
- d. none of the above ☐

1) In binary: $1011 + 11011 = ?$

- a. 101000 ☐
- b. 100110 ☐
- c. 110110 ☐
- d. none of the above ☐

2) Summing up all even integers is a nonalgorithmic problem, because:

- a. It's too difficult to compute ☐
- b. It involves infinite number of operations without termination ☐
- c. No machine can do it ☐
- d. The algorithm to do it will very long ☐

4) The **ALU** is the unit of computer responsible of:

- a. Storing programs and data permanently ☐
- b. Performing arithmetic and logical operations ☐
- c. Performing Input/Output operations ☐
- d. None of the above ☐

5) The value of $((x \leq y) \parallel \text{True})$ is:

- a. $(x > y)$ ☐
- b. $(x \leq y)$ ☐
- c. False ☐
- d. none of the above ☐

6) Von Neumann architecture is:

- a. A stored-program computer ☐
- b. A web browser ☐
- c. An architecture for RAM ☐
- d. All the above ☐

7) $(110)_{10} + (110)_2$ is equivalent to:

- a. $(220)_{10}$ ☐
- b. $(1110100)_2$ ☐
- c. $(1100)_2$ ☐
- d. None of the above ☐

8) The software that controls and manages the computer resources is:

- a. Linker ☐
- b. Loader ☐
- c. Web browser ☐
- d. Operating system ☐

Question 2 (20 points)

Show the output of each of the following program segments:

<pre>bool myfunc(int n); int main () { int num[10] = {7, 9, 11, 15, 17, 19, 21}; for (int k = 0; k < 5; k++) if (myfunc(num[k])) cout << setw(4) << k; cout << endl; } bool myfunc(int n) { bool flag = true; int c = 2; while ((c <= n/2) && (flag)) if (n % c == 0) flag = false; else c++; return flag; }</pre>	
--	--

<pre>float a = 3, b = 2, c = 1.5, x; int d = 4, e = 6.5, y; cout << setiosflags (ios :: fixed ios :: right); x = a * b - c + ++d / e++; y = a * b - c + e / --d; cout << " x = " << setw(4) << x << endl; cout << " y = " << setw(4) << y << endl; a = 3.3456; b = 5.9876; cout << setprecision(2); cout << setw(4) << a << setprecision(1) << setw(4) << b << endl; cout << setprecision(0) << setw(4) << a << setw(4) << b << endl;</pre>	
--	--

<pre>int n = 11101, w = 1, s = 0; const int ten = 10; cout << "The equivalent value of " << n << " = "; do { s = s + w * (n % ten); w = 2 * w; n = n / ten; } while (n != 0); cout << s << endl;</pre>	
--	--

<pre> Void interchange (int&, int&); void main() { int a = 6.5, b = 5.3, c = 3.2; do { if (b < a) interchange (a, b); if (c < b) interchange (b, c); } while ((b < a) (c < b)); cout << "The Final value is:" << endl; cout << setw(4) << (b + c) / 2 << endl; } void interchange (int& X, int& Y) { int T = X; X = Y; Y = T; } </pre>	
---	--

<pre> bool check (int); void main () { int BB [5]; int AB [5] = { 102, 10, 101, 1100, 103}; for (int index = 0; index < 5; index++) if (check (AB[index])) BB[index] = 1; else BB[index] = 0; for (int r = 0; r < 5; r++) cout << setw(3) << BB[r]; cout << endl; system ("pause"); } bool check (int n) { const int ten = 10; bool flag = true; do { if ((n % ten == 0) (n % ten == 1)) n = n / ten; else flag = false; } while ((n != 0) && (flag)); return flag; } </pre>	
--	--

Question 3 (20 points)

1. The following C++ function **rotate** takes three integer numbers over its parameters **a**, **b**, and **c** and then rotates them, such that **a** takes the value of **c**, **b** takes the value of **a**, and **c** takes the value of **b**. The function uses only one additional variable **temp** (as temporary store). After being rotated, the three values should be returned back to the **main** function. Complete the following code of the function **rotate**:

```
..... rotate ( ..... a, ..... b, ..... c)
```

```
{ int temp;
```

...../

.....;

.....; }

2. Write a C++ function **exists** that takes a big positive integer number (**big**) and a single digit (**d**) from **0** to **9**. The function returns **true** if the digit (**d**) does exist in the big positive number (**big**), and returns **false**, otherwise.. (For example: if the function takes **2799817** as a big integer number, and **7** as a single digit, then, it returns **true**, and if the function takes **345086** as a big integer number, and **9** as a single digit, then, it returns **false**. Make your

(Bonus 3 points): Make your function more efficient by terminating it after the first digit (**d**) is found in the big integer number (**big**))

[illegible]

Question 4 (15 points)

Write **only** the C++ nested-loop to produce the following output:

(**Bonus 3 points for only using:** nested-loop + one if statement + one switch statement)

[illegible][illegible]

Question 5 (15 points)

The ancient Greek Euclid developed a method for finding the Greatest Common Divisor (GCD) of two integers **A** and **B** (**A** greater than **B**). His method is:

1. If the remainder (**R**) of **A / B** is 0, then **B** is the **GCD**.
2. If it is not 0, then assign **B** to **A** and the remainder (**R**) to **B**.
3. Return to step 1 and repeat the process.

The following C++ program takes two integer numbers **N** and **M** and prints out their **GCD**. The program is structured as three functions: the **main**, **swap** and **GCD**. The **main** function takes two integer numbers **N** and **M** and if **N < M** it calls the function **swap** that will interchange the values of **N** and **M** (**N** is assigned to **M** and **M** is assigned to **N**). Then, the function **main** calls the function **GCD** that receives the values of **N** and **M** as **A** and **B**, performs Euclid's process (defined above) and returns the corresponding **GCD** value to the **main** function that prints it out.

The C++ program is given below but there are some missings (represented by dots). Complete these missings such that the program could be compiled and run correctly.

```
#include <iostream>
#include <iomanip>
using namespace std;
void swap(int& X, int& Y);
.....
void main ()
{   int N, M;
    cin >> N >> M;
    if (N < M)
        .....

    << cout << "The Greatest Common Divisor of " << setw(4) << N << " and " << setw(4) << M << " is "
    << ..... << endl;

    system ("pause");
}
void swap (.....)
{
    int T = X;
    .....
    .....
}
..... GCD ( ..... A, ..... B )
{
    int R = A % B;
    while ( ..... )
    {   A = B;
        .....
        R = .....
    }

    return .....;
}
```

Question 6 (20 points)

Write a C++ modular program to generate a table showing the equivalent decimal value of a binary value. The table contains 20 rows at most. The program goes through the following steps:

1. The **main** function takes a positive integer number **n** representing the number of rows to be printed in the table. The program validates the entered value for **n** to accept only a value that is greater than **0** but not more than **20**.
2. The main function takes from the user the binary numbers one by one to be stored in an array. Each entered binary number should be validated by sending it to the function **validate_Binary** that checks that it is only a combination of 0's and 1's and returns back an appropriate feedback to the main function.
3. The **main** function prints out the header of the table (see the given example below). Reads the binary numbers one by one from the array and sends it the function **convertBtoD** which computes its equivalent decimal value and returns it back to the **main** function. The program prints a row with the two values; the binary number and its equivalent decimal value.
4. After printing **n** rows, the program prints a footer (see the example below).
5. You are advised to define and use appropriate functions to print the header and footer of the table.

Example: for **n = 5**, and binary numbers = **101, 1101, 111, 1111, and 10001**, the generated conversion table should look like the following one:

```
#####
Binary Number      Decimal Value
-----
          101              5
         1101             13
          111              7
         1111             15
        10001              9
#####
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

[illegible]

Handwriting practice lines consisting of 30 horizontal dotted lines.

Good Luck