

The American University in Cairo
Computer Science & Engineering Department
CSCE 106/1001 "Fundamentals of Computer Science"

Instructor: Dr. Khalil	Final Exam	Spring 2014
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	25	
3	20 + 3	
4	15 + 3	
5	15	
6	15	
Total	100	

Question 1 (10 points)

Tick only one possible answer for each of the following:

1) The Boolean expression $\neg(x > y)$ 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 $((x - 4) \neq 4) \&\& ((x - 4) == 4)$ 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 (25 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; system ("pause"); } 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 = 5.8876; 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 += w * (n % ten); w *= 2; n /= ten; } while (n != 0); cout << s << endl;</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: ";
  cout << setw(4) << (b + c) / 2 << endl;
  system ("pause");
}
void interchange (int& X, int& Y)
{ int T = X;
  X = Y;
  Y = T; }

```

```

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;
}

```

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**.)

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

```
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
```

Question 4 (15 points)

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

(**Bonus 3 points for structuring your solution as only one nested-loop + one if statement + one switch statement**)

```
@ # # # # # # # #
2 3 # # # # # # #
& & & # # # # # #
@ @ @ @ # # # # #
2 3 4 5 6 # # # #
& & & & & # # #
@ @ @ @ @ @ @ # #
2 3 4 5 6 7 8 9 #
& & & & & & &
```

Question 5 (15 points)

A big positive integer number (**BIG**) would be divisible by **9** if the sum of its digits is divisible by 9. For example, **60390045** is divisible by 9 because the sum of its digits **27** is divisible by 9, while **60290018** is not divisible by **9** because the sum of its digits **26** is not divisible by 9.

The following modular C++ program takes a positive integer number (**BIG**) that should be greater than **99999** and not more than **9999999** and notifies the user whether **BIG** is divisible by **9** or not using the above mentioned technique. The program is structured as a **main** function that reads and validates the user input (**BIG**). The **main** function calls another function **divisiblyBy9** that takes the big number, uses the mentioned technique to check it, and returns back the result to the **main**. The **main** function uses the returned result to display the proper message to the user.

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 main ()
{ long BIG;

  cout << "Enter a positive integer greater than 99999 and less than 9999999: " << endl;
  cin >> BIG;
  while (.....)
  { cout << "Invalid Input, Reenter: ";
    cin >> BIG; }
  if ( ..... )
    cout << "The number  " << setw(4) << BIG << " is divisibly by 9" << endl;
  else
    cout << "The number  " << setw(4) << BIG << " is NOT divisible by 9" << endl;
  system ("pause");
} // End of main function

bool divisibleBy9 ( ..... m )
{ int sum = .....;
  while (.....)
  {
    .....
    .....

  }

  return ( ..... );
} // End of divisibleBy9 function
```

Question 6 (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 as follows:

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

Write a modular C++ program that takes two integer numbers **N** and **M** and prints out their **GCD**. The program could be 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.

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

Handwriting practice lines consisting of 30 horizontal dotted lines.

Good Luck