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

Instructor Name: FINAL FALL 2007

Last Name: ...................................................... ID.............................

First Name: ...................................................... SECTION

* 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 Academic Integrity 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**

<table>
<thead>
<tr>
<th>Question</th>
<th>Points</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
1) The value of \((x - 4 != 4) \&\& (x - 4 == 4)\) is:
   a. \(x - 4\) [ ]
   b. True [ ]
   c. \(x + 4\) [ ]
   d. None of the above [ ]

2) The smallest integer number that can be stored in 16-bit unsigned format is:
   a. 0 [ ]
   b. -128 [ ]
   c. 127 [ ]
   d. None of the above [ ]

3) The largest integer number that can be stored in 8-bit two’s complement is:
   a. 128 [ ]
   b. 127 [ ]
   c. 64 [ ]
   d. None of the above [ ]

4) \((90E)_{16}\) is equivalent to:
   a. \((9010)_{10}\) [ ]
   b. \((6777)_{8}\) [ ]
   c. \((100100001110)_{2}\) [ ]
   d. None of the above [ ]

5) \((1010001111)_{2}\) is equivalent to:
   a. \((507)_{8}\) [ ]
   b. \((507)_{16}\) [ ]
   c. \((5544)_{10}\) [ ]
   d. None of the above [ ]

6) \((-15)_{10}\) in 8-bit two’s complement format is:
   a. 110001 [ ]
   b. 11110000 [ ]
   c. 11110001 [ ]
   d. None of the above [ ]

7) Von Neumann architecture is:
   a. A stored-program computer [ ]
   b. A web browser [ ]
   c. An architecture for RAM [ ]
   d. All the above [ ]

8) The Assembler is a software that:
   a. translates a high-level language program into machine language [ ]
   b. translates an assembly language program into machine language [ ]
   c. lets users display and view a web document [ ]
   d. None of the above [ ]

9) The value of \(10 \% 4 * 3 - 8 >= 18 + 30 / 4 - 20) \&\& (7 == 7)\) is:
   a. -2 [ ]
   b. True [ ]
   c. False [ ]
   d. None of the above [ ]

10) Software Piracy is:
    a. Violation of copyright agreements by illegally copying software [ ]
    b. Code disrupting the computer operations and erasing information [ ]
    c. A Computer theft or computer fraud [ ]
    d. A biological disease causing malfunctioning of the computer [ ]

11) The RAM is used to:
    a. To store fixed code to start up the computer [ ]
    b. To store programs and data that are to be executed [ ]
    c. To perform the arithmetic and logical operations [ ]
    d. None of the above [ ]
12) In binary: 1011 + 11011 = ?
   a. 101000 [   ]
   b. 100110 [   ]
   c. 110110 [   ]
   d. 100101 [   ]

13) The representation of -0 in 16-bit Two’s complement is:
   a. 1000000000000000 [   ]
   b. 1111111111111111 [   ]
   c. 0000000000000001 [   ]
   d. None of the above [   ]

14) 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 [   ]

15) The Compiler is a software that:
   a. translates a high-level language program into machine language [   ]
   b. translates an assembly language program into machine language [   ]
   c. lets users display and view a web document [   ]
   d. None of the above [   ]

**Question 2 (10 points)**
Write a program fragment that uses nested loops to produce the following output:

```
1  2  3  4  5  6
0  2  3  4  5  6
0  0  3  4  5  6
0  0  0  4  5  6
0  0  0  0  5  6
0  0  0  0  0  6
```
Question 3 (25 points)
Show the output of the following program segments:

<table>
<thead>
<tr>
<th>Program</th>
<th>Output</th>
</tr>
</thead>
</table>
| `# include <iostream>`<br>`# include <iomanip>`<br>`using namespace std;`<br>`void main()`<br>`{`<br>`    int num = 56067, d, s = 0;`<br>`    const int ten = 10;`<br>`    d = num % ten;`<br>`    s = s + d;`<br>`    num = num / ten;`<br>`    d = num % ten;`<br>`    s = s + d;`<br>`    num = num / ten;`<br>`    d = num % ten;`<br>`    s = s + d;`<br>`    d = num / ten;`<br>`    s = s + d;`<br>`    cout << "The result is " << setw(4) << s << endl;`<br>`}`<br>`int A[8] = {37,59,25,35,25,44,75,65};`<br>`int T;`<br>`int n = 5;`<br>`for (int j = 0; j < n; ++j)`<br>`    if (A[j] > A[j+1])`<br>`    `{`<br>`        T = A[j];`<br>`        A[j] = A[j+1];`<br>`        A[j+1] = T;`<br>`    }`<br>`int myfunc(int n);`<br>`void main()`<br>`{`<br>`    for (int k=0; k <= 6; k += 2)`<br>`        cout << setw(3) << k << setw(5) << myfunc(k) << endl;`<br>`    cout << "End of processing" << endl;`<br>`}`<br>`int myfunc(int n)`<br>`{`<br>`    int f = 1;`<br>`    for (int m=2; m <= n; m++)`<br>`        f *= m;`<br>`    return f;`<br>`}
```cpp
#include <iostream>
using namespace std;

void Myout(int);

void main()
{
    for (int num = 5; num <= 15; num += 4)
        Myout(num);
    cout << "End of Processing" << endl;
}

void Myout (int n)
{
    int B[ ] = {0, 0, 0, 0};
    int m = n, k = 0;
    do
    {
        B [k] = m % 2;
        m = m / 10;
        k++;
    } while ( m > 0 );
    cout << n << "         ";
    for (k = 3; k >= 0; k--)
        cout << B[k];
    cout << endl;
}

#include <iostream>
#include <iomanip>
using namespace std;
void test ( int, int&, int&);
void main()
{
    int d = 15, e = 17 , f = 13 , a = 19 , b = 21;
    test (d, e, f);
    cout << "In the main program after the first call, ";
    cout << "the variables are: " << setw(3) << e
         << setw(3) << d << setw(4) << f << endl;
    cout << "------------------------------------------------------------
    -------" << endl;
    test (a, b, f);
    cout << "In the main program after the second call, ";
    cout << "the variables are: " << setw(3) << a
         << setw(3) << b << setw(4) << f << endl;
}
void test(int t, int& s, int& x)
{
    s = 4;
    s = s - 3;
    x %= (++s + ++t);
    t = 7 * s++;
    cout << "Function test Output: " << setw(3) << s
         << setw(3) << t << setw(3) << x << endl;
}
```
Question 4 (15 points)

A Prime number is a positive integer number greater than 1 which is only divisible by itself (and eventually by 1). The first prime number is 2, and the next is 3, and so on.

The following C++ program lists all the prime numbers in the range of 2 to 1000. There are some missings (represented by dots) in the given program. Complete these missings such that the program could be compiled and run correctly.

The Program

```cpp
#include <iostream>
using namespace std;

............ prime (........);

void main ()
{
  for (int num = 2; .................; ...........)
    if (........................................)
      cout << num << endl;
}

bool prime ( int n)
{
  bool flag = true;
  int c = 2;
  while (..........................................................)
    if (..........................................................)
      flag = false;
    else
      ..........................................................;
      ..........................................................
  }
```
Question 5 (15 points)
Write a C++ program that takes as an input a decimal positive integer number not greater than 255 and displays its equivalent binary representation on the screen in 8-bit format. For example, if the input is 25, the binary 00011001 should be displayed, and if the input is 98, the binary 01100010 is displayed, and if the input is 0, the binary 00000000 should be displayed, and so on.

Enforce validation on the input such that the program should not accept any integer number less than 0 or greater than 255.

The program

```cpp
// Example code for the program
```
Question 6 (20 points)
Write a modular C++ program that allows student of AUC to compute his/her GPA. The student has to enter to the program the following:

- The number of courses the student has taken.
- For each course, the letter grade the student has received in the course (the program should not accept any value for the letter grade other than A, B, C, D, and F) and the course credit hours (the program should not accept any value for the course credit hours other than 1, 2, and 3).

The numerical value of each letter grade is determined according to the following table:

<table>
<thead>
<tr>
<th>Letter grade</th>
<th>Numerical value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>4.0</td>
</tr>
<tr>
<td>B</td>
<td>3.0</td>
</tr>
<tr>
<td>C</td>
<td>2.0</td>
</tr>
<tr>
<td>D</td>
<td>1.0</td>
</tr>
<tr>
<td>F</td>
<td>0.0</td>
</tr>
</tbody>
</table>

The formula for computing the GPA is as follows:

\[
GPA = \frac{\text{SumOf} \ (\text{gradeNumericValue} \times \text{CourseCreditHours})}{\text{SumOf} \ (\text{CourseCreditHours})}
\]

For example, if the student enters 3 as a number of courses in which he/she got the letter grades B, A, and C and these courses are 3, 2, 3 credit hours respectively, the GPA is computed as follows:

\[
GPA = \frac{3.0 \times 3 + 4.0 \times 2 + 2.0 \times 3}{3 + 2 + 3}
\]

You may structure your program as follows:

- The main function reads the number of courses.
- For each course, the main function calls the function validCrHr that reads and returns a valid number of credit hours (1, 2, or 3) for that course. For the same course, the main function calls another function validGrade that reads a valid letter grade of the course (A, B, C, D, or F) and returns its equivalent numeric value.

**The Program**

```cpp

```

```cpp
```

Page 8 of 9