The American University in Cairo
Computer Science Department
CSCI 106

Instructor: …………..     FINAL EXAM     Spring 2007

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

First Name: .............................................     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 Academic Integrity 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>10</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>30</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>
**Question 1 (10 points)**

Tick **only one possible** answer for each of the following:

1) The software that controls and manages the computer resources is:
   a. Source program [ ]
   b. Loader [ ]
   c. Web browser [ ]
   d. Operating system [ ]

2) The smallest integer number that can be stored in 8-bit unsigned format is:
   a. -0 [ ]
   b. -128 [ ]
   c. 127 [ ]
   d. None of the above [ ]

3) In binary: 1011 + 11011 = ?
   a. 101000 [ ]
   b. 100110 [ ]
   c. 110110 [ ]
   d. 100101 [ ]

4) The largest integer number that can be stored in 8-bit two’s complement is:
   a. 128 [ ]
   b. 1127 [ ]
   c. 64 [ ]
   d. None of the above [ ]

5) \((10E)_{16}\) is equivalent to:
   a. \((9010)_{10}\) [ ]
   b. \((6777)_{8}\) [ ]
   c. \((100001110)_{2}\) [ ]
   d. None of the above [ ]

6) \((101000011)_{2}\) is equivalent to:
   a. \((503)_{8}\) [ ]
   b. \((503)_{16}\) [ ]
   c. \((554)_{10}\) [ ]
   d. None of the above [ ]

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

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

9) A web browser is:
   a. A compiler that translates a source program to an object code [ ]
   b. A graphical user interface that allows users to navigate through the web [ ]
   c. A software that lets users edit documents [ ]
   d. A computer that provides resources to other computers in a network [ ]

10) 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 [ ]
**Question 2 (10 points)**

A result of \((-128)_{10}\) was produced in an accumulator of 8-bit structure. Give the binary content of the accumulator for each of the following cases:

1. The computer is performing its operations in one’s complement integer format:

```
```

2. The computer is performing its operations in two’s complement integer format:

```
```

3. The computer is performing its operations in sign-magnitude integer format:

```
```

**Question 3 (30 points)**

Show the output of each of the following program segments:

---

<table>
<thead>
<tr>
<th>Program - 1</th>
</tr>
</thead>
</table>
| int x = 1, y = 1;  
while (x < 4)  
{    y = y * x++;  
cout <<"x = "<<setw(3)<<<x<<"y = "<<setw(3)<<y<<endl;  
} |

<table>
<thead>
<tr>
<th>Program - 2</th>
</tr>
</thead>
</table>
| int F[6] = {1, 1};  
const int zero = 0;  
const int one = 1;  
cout << setw(3) << zero << setw(3) << F[0] << endl;  
cout << setw(3) << one << setw(3) << F[1] << endl;  
for (int k = 2; k < 6; k++)  
{    F[k] = F[k-1] + F[k-2];  
cout << setw(3) << k << setw(3) << F[k] << endl;  
} |

<table>
<thead>
<tr>
<th>Program - 3</th>
</tr>
</thead>
</table>
| const int ten = 10;  
int d;  
int n = 7856;  
do  
{    d = n % 10;  
cout << d;  
n /= ten;  
} while (n != 0); |
void nmx (int, int, float&);

void main()
{
    int n = 3;
    int m = 2;
    float nm = 2.757;
    nmx (n, m, nm);
    cout << "n = " << setw(3) << n << endl;
    cout << "m = " << setw(3) << m << endl;
    cout << setprecision(2);
    cout << "result is:   " << setw(6) << nm << endl;
}

void nmx (int a, int b, float& x)
{
    a /= b;
    if (a % b != 0)
    {
        x = 4.75 + a;
    }
    else
    {
        x = 4.75 - b;
    }
}

const int n = 6;
int A[n] = {4, 4, 2, 5, 3, 9};
bool flag = true;
int k = 0;
while ( (k < n - 1) && (flag) )
{
    if (A[k] > A[k+1])
    {
        flag = false;
        cout << setw(2) << k << setw(2) << A[k] << endl;
    }
    else
    {
        k++;
    }
}
if (flag)
{
    cout << " The list is OK" << endl;
}

int x = 1, y = 1;
while (x < 3)
{
    y = y * ++x;
    cout << x << "x = " << setw(3) << x << endl;
    cout << y << "y = " << setw(3) << y << endl;
}
Question 4 (15 points)
A graduating student of AUC is awarded an honorary degree according to his/her final GPA. The honorary degree is granted according to the following rules:

<table>
<thead>
<tr>
<th>Final GPA</th>
<th>Honorary Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.80 &lt;= GPA &lt;= 4.00</td>
<td>Highest Honors</td>
</tr>
<tr>
<td>3.60 &lt;= GPA &lt; 3.80</td>
<td>High Honors</td>
</tr>
<tr>
<td>3.40 &lt;= GPA &lt; 3.60</td>
<td>Honors</td>
</tr>
<tr>
<td>2.00 &lt;= GPA &lt; 3.40</td>
<td>Pass with no Honors</td>
</tr>
</tbody>
</table>

Draw a flow chart and write a program in C++ to input the GPA of a student and print out his/her honorary degree. **Show the three phases of software development: the analysis, design, and implementation. Implement your solution in C++ once using nested-if structure and second using switch structure.**

Enforce validation on the input GPA such that it is not less than 2.00 and not greater than 4.00.
The Program Using nested-if structure

The Program Using switch structure
**Question 5 (15 points)**

One interesting application of computers is the drawing of graphs and bar charts (sometimes called “histograms”). Write a modular program that uses an iterative structure to read ten positive integer numbers (each between 1 and 30). For each number read, your program should print a line containing the read number (on two positions), followed by a space and then that number of adjacent ‘#’. For example, if your program reads 5, 10, 2, 7, 7, 15, 9, 3, 12, and 2 it should print:

```
5  #####
10  #############
  ##
  #####
  #######
15  ##################################################
  #######
  #######
  #############
3  #######
12  #########################################
  ##
2  #######
```

Define a function `drawLine` that takes as an input an integer number and prints out its line. (the integer number itself on two positions, followed by a space and then its corresponding number of ‘#’.
**Question 6 (20 points)**

Write a modular program in C++ to generate a table converting temperatures from Celsius to Fahrenheit (look at the example below). The program should perform the following:

1. Reading two integer numbers $M$ and $N$ defining the range of Celsius temperatures (look at the example below in which $M = 50$ and $N = -10$) through calling the function `inputRange` which should validate that $N$ should be less than $M$ and and in case $N$ is not less than $M$ the function `inputRange` should call another function `swap` that takes the two integer numbers $M$ and $N$ and swap them (interchange their values).

2. Reading the step value $S$ which is used as a decrement to move from one Celsius temperature to the next (look at the example shown below in which the step $S = 5$) through calling the function `inputStep` which should accept only positive integer greater than 0 and not greater than the difference between $M$ and $N$.

3. For each Celsius temperature, the main function should compute its equivalent Fahrenheit $(\text{Fahrenheit} = 1.8 \cdot \text{Celsius} + 32.0)$ through calling the function `computeFahrenheit`.

4. Printing out the table of conversion according to the input values (range and step) using appropriate header and format. **The computed Fahrenheit temperature should be printed out rounded (approximated) to the nearest integer value.**

An Example: $M$ (start value) = 50, $N$ (end value) = -10, and $S$ (step) = 5

<table>
<thead>
<tr>
<th>Celsius</th>
<th>Fahrenheit</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>122</td>
</tr>
<tr>
<td>45</td>
<td>113</td>
</tr>
<tr>
<td>40</td>
<td>104</td>
</tr>
<tr>
<td>...</td>
<td>23</td>
</tr>
<tr>
<td>-5</td>
<td>14</td>
</tr>
</tbody>
</table>

---
Good luck