Do not turn this page until asked to do so.

* Exam time is 60 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>20</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>100</strong></td>
<td></td>
</tr>
</tbody>
</table>
Question 1 (20 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 2 (20 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++ 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 Analysis**

---

**The Flow Chart**

---
Question 3 (15 points)
A Perfect integer number is a positive integer number greater than 1 and whose sum of its factors (including 1) is equal to the number itself (for example, 6 is the first perfect number because \(1 + 2 + 3 = 6\)).

The following C++ program takes a positive integer number greater than 1 and less than or equal 2000 and prompts the user whether the entered number is perfect or not perfect. 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 includes validation on user’s input to accept an integer greater than 1 and less than or equal 2000.

**The Program**

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

void main ()
{
    int Mynumber, total;

    do
    {
        cout << "Enter a positive integer greater than 1 and less than or equal 2000: " << endl;
        cin >> Mynumber;
    } while (..........................);

    for ( int k = 1 ; ..................; k++)
    {
        if ( .........................)
            total += .........;
        if ( .......................... )
            cout << "The number " << setw(4) << Mynumber << " is Perfect" << endl;
        else
            cout << "The number " << setw(4) << Mynumber << " is Not Perfect" << endl;
    }
```
### Question 4 (20 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 = 2;`  
| while (x < 3)  
| y /= ++x;  
| cout << "x = " << setw(3) << x << " y = " << y << endl;`  |

<table>
<thead>
<tr>
<th>Program - 2</th>
</tr>
</thead>
</table>
| `const int ten = 10;`  
| `int d;`  
| `int n = 735;`  
| `do`  
| `{`  
| `d = n % 10;`  
| `cout << d;`  
| `n /= ten;`  
| `}`  
| `while (n != 0);`  |

<table>
<thead>
<tr>
<th>Program - 3</th>
</tr>
</thead>
</table>
| `int x, y = 40;`  
| `for (x = 5; x > y; x *= 2)`  
| `cout << setw(3) << x << setw(3) << y << endl;`  
| `cout << "The Final Value Is: << setw(5) << x / 2 % y;"`  |

<table>
<thead>
<tr>
<th>Program - 4</th>
</tr>
</thead>
</table>
| `int x = 2, y = 2;`  
| `while (x < 4)`  
| `{`  
| `y *= x++;`  
| `cout << "x = " << setw(3) << x << " y = " << setw(3) << y << endl;}`  |

---

Page 6 of 7
Question 5 (25 points)
A positive integer number is divisible by 9 if the sum of its digits is also divisible by 9. Using this technique, write a C++ program that takes a positive integer number with any number of digits but less than or equal to 99999 and notify the user whether the number is divisible by 9 or not. Include in your program the necessary validation on the user’s input.

The Program

Good Luck