EXAMINATION INSTRUCTIONS

* 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>10</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>50</td>
<td></td>
</tr>
</tbody>
</table>
Question 1 (10 points)

Write a program fragment that uses nested loops to produce the following output:

```
*#
*###
*###*
*###**#
*###***#
*###****#
*###*****#
*###******#
*###*******#
*###********#
*###*********
```

Question 2 (5 points)

1. Rewrite the Boolean expression eliminating the not (!) operator.

   ```plaintext
   ! ( x < y )
   ! ( ( x <= y ) || ( s > t ) )
   ```

2. What is the value of the following expressions:

   ```plaintext
   (x == x)
   (x != 7) && (x == 7)
   (y != 5) || (y == 5)
   ```
Question 3 (15 points)
Show the output of each of the following program segments:

```cpp
int F[6];
const int zero = 0;
const int one = 1;
F[zero] = 0;
F[one] = 1;
cout << setw(3) << zero << setw(3) << F[zero] << endl;
cout << setw(3) << one << setw(3) << F[one] << 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;
}
```

```cpp
int x = 1, y = 1;
while (y < 4)
{
    y = y + y / x++;
cout << setw(3) << x << setw(3) << y << endl;
}
```

```cpp
int k;
for (k=1; k <= 20; k++)
    if ((k % 9) == 0) && (k % 3) != 0)
        cout << k << endl;
```

```cpp
int x = 1, y = 1;
do
{
    y += y / ++x;
cout << setw(3) << x << setw(3) << y << endl;
} while (y > 1);
```

```cpp
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(4) << x / 2 % y;
```

**Question 4 (20 points)**

Write a program in C++ to determine and store all the prime numbers in a given range that is defined by two positive integers \( n \) and \( m \). The prime numbers should be stored in an array \texttt{prime}. \( n \) should be a positive integer greater than 2 and marks the start value of the range. \( m \) should be a positive integer greater than \( n \) and marks the end value of the range. For example, if the user enters 10 for \( n \) and 100 for \( m \), the program should determine and store (in the array \texttt{prime}) all the prime numbers in the range from 10 to 100, inclusive.

The program should accept from the user a value for \( n \) and a value for \( m \) after validating each value according to the above given conditions (\( n \) should be positive integer greater than 2 and \( m \) should be positive integer greater than \( n \)).

It is also required to print out the smallest and largest numbers that are found in the given range.

**The Program**

```cpp
// Insert program code here
```