CSC1005: Introduction to Computer Engineering

Programming and Applications

Assignment 1

Assignment description:

This assignment will be worth 7% of the final grade.

You should write your code for each question in a .py file (please name it using the question name, e.g. q1.py). Please pack all your .py files into a single .zip file, name it using your student ID (e.g. if your student ID is 123456, then the file should be named as 123456.zip), and then submit the .zip file via Blackboard.

Please create a Word document that provides a detailed explanation of the code for each question. This document should clearly outline the logic behind your code, specify the expected inputs, and describe the anticipated outputs. Make sure the explanations are easy to understand and provide enough context for anyone reviewing your work. Include this Word document in the .zip file along with your code.

Please be aware that the teaching assistant may ask you to explain your code to verify that it was written by you. Additionally, your submission may be checked for similarities with other students' work using Blackboard to ensure academic integrity.

This assignment is due on 5:00PM, 17 October (Friday). For each day of late submission, you will lose 10% of your mark in this assignment. If you submit more than three days later than the deadline, you will receive zero in this assignment.

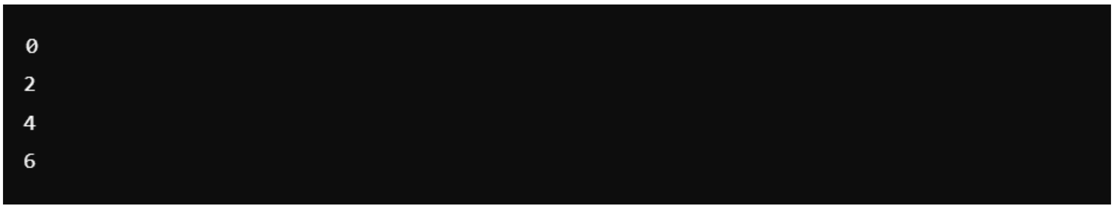
If you have any questions about Assignment 1 (AS1), please contact the TA for AS1, MENG Xiangrui, at 222010046@link.cuhk.edu.cn

Question 1 (10% of this assignment):

Write a program that prompts the user to input an integer n, and then uses a for loop to print the first n even numbers.

Example:

If the user enters 4, the program should output:

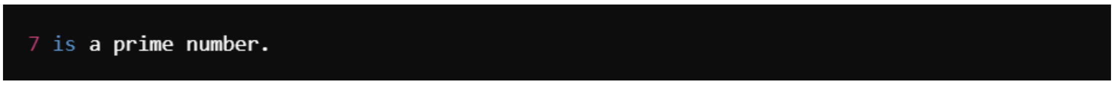


```
0
2
4
6
```

Question 2(10% of this assignment):
Write a program that asks the user to input a positive integer n. The program should then print whether n is prime or not.

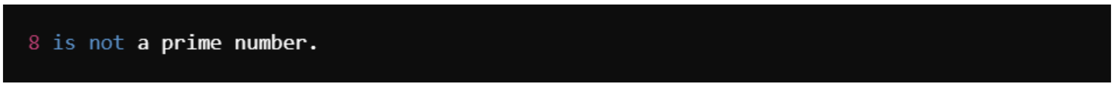
A prime number is a number greater than 1 that has no divisors other than 1 and itself.

Example:
If the user enters 7, the program should output:



```
7 is a prime number.
```

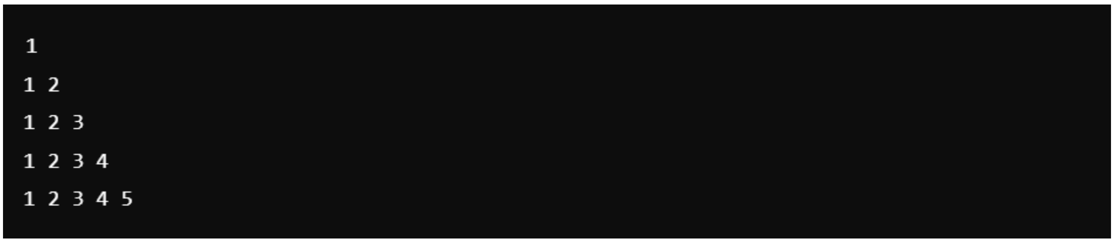
If the user enters 8, the program should output:



```
8 is not a prime number.
```

Question 3 (15% of this assignment):
Write a program that allows the user to input an integer N and prints a pattern of N rows of numbers, where each row starts from 1 and increases up to the row number.

Example:
If the user inputs N = 5, the program should output:



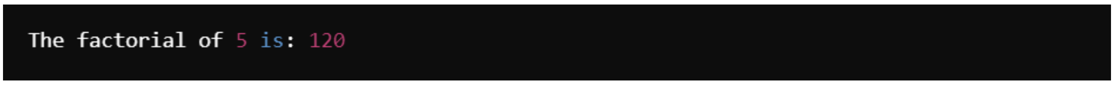
```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Ensure that your program handles improper inputs (e.g., if the user enters a non-positive integer).

Question 4 (15% of this assignment):
Write a program that allows the user to input an integer N and prints the factorial of N. The factorial of a number is the product of all positive integers from 1 up to N. For example, the factorial of 5 (denoted 5!) is 1 * 2 * 3 * 4 * 5 = 120.

Make sure your program is robust enough to handle improper inputs (e.g., if the user enters a negative number).

Example:
If the user inputs N = 5, your program should output:



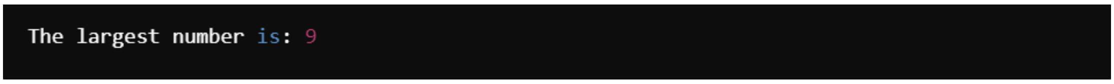
```
The factorial of 5 is: 120
```

Question 5(20% of this assignment):

Write a program that prompts the user to input a list of integers separated by spaces. Your program should then find the largest integer in the list and print it.

Example:

If the user inputs 3 5 2 9 6, the program should output:



```
The largest number is: 9
```

Question 6 (30% of this assignment):

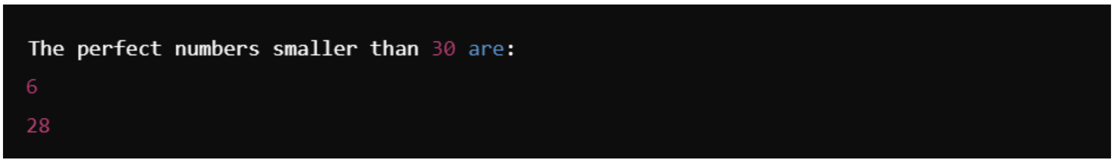
Write a program that allows the user to input a positive integer N and prints a list of all **perfect numbers** smaller than N. A **perfect number** is a positive integer that is equal to the sum of its proper divisors (excluding itself). For example, 6 is a perfect number because its divisors (excluding itself) are 1, 2, and 3, and 1 + 2 + 3 = 6.

Your program should output each perfect number found in a new line.

Additionally, your program should handle improper inputs (e.g., if the user enters a negative number or a non-integer value).

Example:

If the user inputs N = 30, the program should output:



```
The perfect numbers smaller than 30 are:
6
28
```