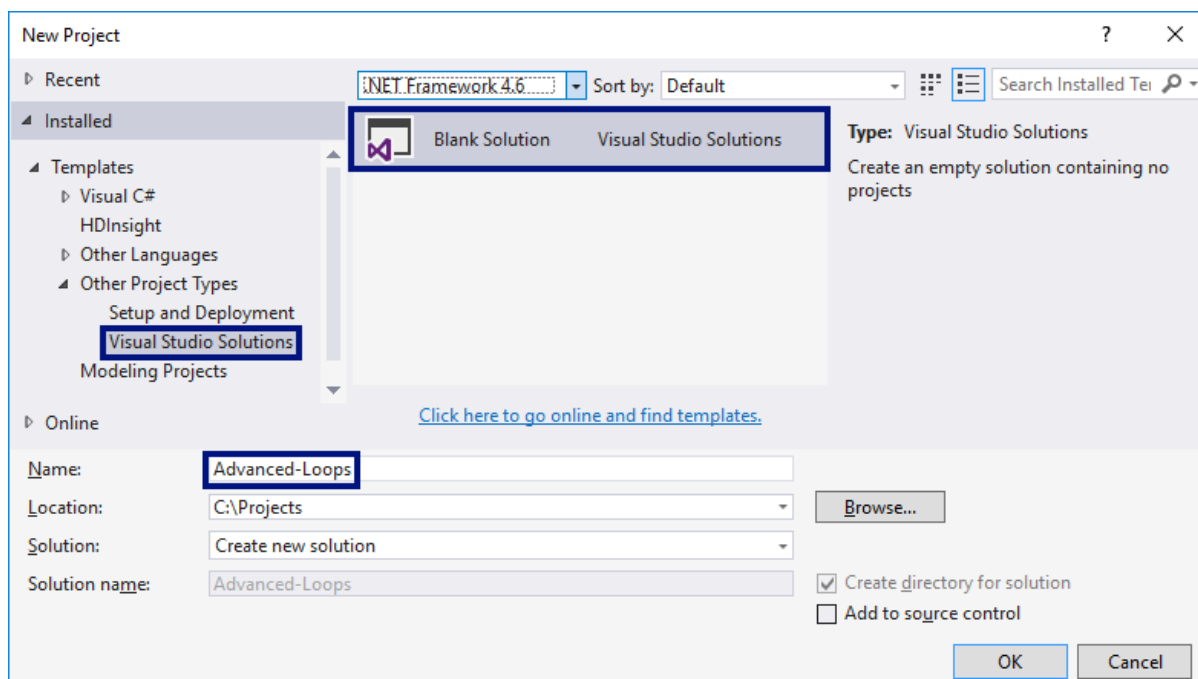




Упражнения: Повторения от по-висока сложност

0. Празно Visual Studio решение (Blank Solution)

Създайте празно решение (Blank Solution) във Visual Studio за да организирате кода от задачите за упражнение. Целта на този blank solution е да съдържа по един проект за всяка задача от упражненията.



Задайте да се стартира по подразбиране текущия проект (не първият в решението). Кликнете с десен бутон на мишката върху Solution 'Advanced-Loops' → [Set StartUp Projects...] → [Current selection].

1. Факториел

Напишете програма, която чете цяло число n ($1 \leq n \leq 12$), въведено от потребителя, и изчислява и отпечатва $n! = 1 * 2 * \dots * n$ (n факториел). Примери:

ВХОД	ИЗХОД
5	120

ВХОД	ИЗХОД
6	720

ВХОД	ИЗХОД
10	3628800

ВХОД	ИЗХОД
1	1

ВХОД	ИЗХОД
2	2

Подсказка: в цикъл умножете числата от 1 до n .

2. Сумиране на цифрите на число

Напишете програма, която чете цяло число num , въведено от потребителя, и отпечатва сумата от цифрите му.

Примери:

ВХОД	ИЗХОД
5634	18

ВХОД	ИЗХОД
19	10

ВХОД	ИЗХОД
5	5

ВХОД	ИЗХОД
17151	15



Подсказка: в цикъл докато не стигнете до 0 сумирайте последната цифра на числото ($\text{num \% } 10$) и го разделяйте след това на 10 (така изтривате последната му цифра).

3. Проверка за просто число

Напишете програма, която чете цяло число n , въведено от потребителя, и проверява дали е просто число (дали се дели само на себе си и на единица). Да се отпечата "Prime" или "Not prime". Примери:

ВХОД	ИЗХОД
2	Prime

ВХОД	ИЗХОД
3	Prime

ВХОД	ИЗХОД
4	Not Prime

ВХОД	ИЗХОД
5	Prime

ВХОД	ИЗХОД
7	Prime

ВХОД	ИЗХОД
1	Not Prime

ВХОД	ИЗХОД
0	Not Prime

ВХОД	ИЗХОД
-1	Not Prime

ВХОД	ИЗХОД
149	Prime

Подсказки:

- Ако числото е по-малко от 2, значи не е просто.
- Всяко друго число първоначално се приема за просто и се проверява в цикъл дали се дели на числата 2, 3, 4, ..., до корен квадратен от n (цялата част).

4. Въвеждане на четно число (с обработка на грешен вход)

Напишете програма, която чете четно число, въведено от потребителя. Ако потребителят въведе грешно число (нечетно число или стринг, който не е цяло число), трябва да му излиза съобщение за грешка и да въвежда отново. Примери:

ВХОД / ИЗХОД
Enter even number: 34 Even number entered: 34
Enter even number: 35 The number is not even. Enter even number: hello Invalid number! Enter even number: 12.85 Invalid number! Enter even number: 3464232636536513 Invalid number! Enter even number: 8 Even number entered: 8

Подсказки:

- В цикъл въвеждайте число и проверявайте дали е четно. При коректно число излезте от цикъла.
- С `try { ... } catch { ... }` конструкция прихващане грешните числа, които не могат да се обърнат в `int`.



5. Числа на Фибоначи

Напишете програма, която чете цяло число n , въведено от потребителя, и пресмята n -тото число на Фибоначи. Нулевото число на Фибоначи е 1, първото е също 1, а всяко следващо е сумата от предходните две. Примери:

ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД
0	1	1	1	2	2	5	8	10	89

Подсказка:

- При $n < 2$ отпечатайте 1.
- Започнете от $f0=1$ и $f1=1$ и в цикъл сумирайте последните две числа. Записвайте последните две числа след всяка стъпка в $f0$ и $f1$.

6. Пирамида от числа

Напишете програма, която чете цяло число n , въведено от потребителя, и отпечатва пирамида от числа като в примерите:

ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД
7	1 2 3 4 5 6 7	10	1 2 3 4 5 6 7 8 9 10	12	1 2 3 4 5 6 7 8 9 10 11 12	15	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Подсказка:

- С два вложени цикъла печатайте пирамида от числа: на първия ред едно число, на втория ред 2 числа, на третия ред 3 числа и т.н.
- В отделен брояч пазете колко числа сте отпечатали до момента (и кое е текущото число). Когато стигнете n , излезте внимателно от двата вложени цикъла с **break** или **return**.

7. Таблица с числа

Напишете програма, която чете цяло число n , въведено от потребителя, и отпечатва таблица (матрица) от числа като в примерите:

ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД
2	1 2 2 1	3	1 2 3 2 3 2 3 2 1	4	1 2 3 4 2 3 4 3 3 4 3 2 4 3 2 1	5	1 2 3 4 5 2 3 4 5 4 3 4 5 4 3 4 5 4 3 2 5 4 3 2 1

Подсказка:

- С два вложени цикъла за row (ред) и col (колона) печатайте число по формулата $num = row + col + 1$.
- За долната дясна половина на таблицата ще се получат грешни резултати. Там използвайте формулата $2*n - num$.



- Как се сетихме за тези формули? Математическа досетливост: наблюдаваме числата, предполагаме каква е формулата, тестваме и ако не се получи, измисляме друга формула и пак пробваме. В случая имаме различни формули за горната лява и долната дясна половина от матрицата.