**Варіант 1,11**

На гоночній трасі у вас є 6 команд та 7 машин. Кожна команда починає гонку з машиною, і одна машина зупиняється на піт-стопі. Користувач повинен керувати цими машинами та їх командами таким чином:

1. На старті кожній команді користувач надається машина.

2. Одна з машин залишається на піт-стопі та чекає подальших інструкцій.

3. Користувач може вибрати машину, що знаходиться на піт-стопі, та присвоїти їй номер однієї з команд. Машина, яка була у цій команді, переходить на піт-стоп.

4. Машина, що зупинилася на піт-стопі, чекає на нові інструкції.

5. Обмін машинами між командами може відбуватися доти, доки користувач не перерве виконання програми.

6. Машина на піт-стопі може бути відправлена на ремонт, при цьому вона недоступна для обміну.

7. Користувач також бачить таку інформацію щодо кожної команди:

– Час початку руху на машині цієї команди.

- кількість піт-стопів, які зробила команди тобто. змінила машину.

– Номер машини, на якій зараз рухається команда.

Завдання користувача - ефективно керувати машинами та командами, приймаючи рішення про переходи машин між командами, відправлення на ремонт та відстеження статистики по кожній команді.

**Варіант 2.,12**

Створіть консольну програму, яка дозволить користувачеві конвертувати суму з однієї валюти в іншу з використанням заданих курсів обміну.

Користувач вводить суму у вихідній валюті (наприклад, долари).

Користувач вибирає цільову валюту, у яку хоче конвертувати суму (наприклад, євро).

Користувач вводить суму у вихідній валюті.

Додаток використовує задані курси обміну та здійснює конвертацію у вибрану цільову валюту.

Результат конвертації відображається на екрані.

Користувачеві пропонується можливість провести нову конвертацію або завершити роботу програми.

**Варіант 3,13**

Гра "Вгадай число" передбачає, що комп'ютер загадує випадкове число у певному діапазоні, а гравець повинен вгадати це число, отримуючи підказки від комп'ютера.

Комп'ютер випадково вибирає задумане число в заданому діапазоні (наприклад, від 1 до 100).

Гравцеві дається обмежена кількість спроб (наприклад, 10).

Гравець вводить свою гіпотезу (передбачуване число) з клавіатури.

Комп'ютер перевіряє, чи введене число є загадковим числом:

Якщо число збігається із загаданим, гравець переміг і гра завершується.

Якщо число більше за загадане, гравець отримує підказку "Загадане число менше".

Якщо число менше загаданого, гравець отримує підказку "Загадане число більше".

Гравець продовжує вводити здогади і отримувати підказки до тих пір, поки не вгадає число чи закінчиться кількість спроб.

Якщо кількість спроб вичерпано, гравцеві повідомляється, що він програв і гра завершується.

**Варіант 4,14**

Створіть консольну програму на C#, яка дозволить користувачеві вводити два тексти, а потім перевіряє, чи ці тексти є анаграмами.

Користувач вводить перший текст.

Користувач вводить другий текст.

Програма аналізує обидва тексти та перевіряє, чи є вони анаграмами один одного.

Результат перевірки відображається на екрані. Якщо тексти є анаграмами, програма повідомляє про це, інакше повідомляє, що тексти не є анаграмами.

Користувачеві пропонується перевірити інші тексти або завершити роботу програми.

**Варіант 5,15**

Створіть консольну програму, яка дозволить клієнтам вибирати вільні номери із загального списку. Надається 10 номерів, і клієнти можуть вибирати номер, якщо він доступний.

Створіть масив або список із 10 номерів (наприклад, номерів готельних кімнат), всі номери вільні.

Клієнти можуть вводити номери, які хочуть забронювати.

Програма перевіряє, чи вибраний номер доступний. Якщо номер доступний, він бронюється і клієнт отримує підтвердження. Якщо номер зайнятий, клієнт отримує повідомлення про зайнятість номера.

Стан усіх номерів (зайнятий або вільний) відображається на екрані.

Користувачі можуть продовжувати вибирати номери, поки всі номери не будуть зайняті.

За запитом користувача можна відобразити список усіх доступних номерів або завершити роботу програми.

**Варіант 6,16**

Створіть консольну версію класичної гри "Хрестики-нуліки". Гравці по черзі ставитимуть "хрестики" та "нуліки" на ігровому полі. Гра перевіряє наявність переможця після кожного ходу.

Створіть ігрове поле розміром 3x3, де гравці можуть розміщувати "хрестики" (X) та "нуліки" (O).

Перший гравець (наприклад, Х) починає гру.

Гравці по черзі вводять координати комірки (рядок та стовпець) для розміщення свого символу.

Гра триває доти, доки не буде оголошено переможця або відбудеться нічия.

Переможець визначається, якщо він має 3 своїх символи в ряд по вертикалі, горизонталі або діагоналі.

Якщо всі осередки заповнені, і немає переможця, оголошується нічия.

Після завершення гри надається можливість розпочати нову гру або завершити роботу програми.

**Варіант 7,17**

Створіть консольну програму, яка дозволяє користувачеві генерувати випадкові паролі з параметрами, що настроюються: довжина пароля, використання символів, цифр, великих і малих літер.

Користувач вводить параметри для генерації пароля:

Довжина пароля.

Увімкнення/вимкнення символів (наприклад , ! @#$).

Увімкнення/вимкнення цифр.

Увімкнення/вимкнення великих букв.

Увімкнення/вимкнення малих літер.

Програма генерує випадковий пароль на основі вибраних параметрів та виводить його на екран.

Користувач може повторно генерувати паролі з різними параметрами.

Користувач може завершити роботу програми.

Чудово, ось докладне завдання для створення симулятора випадкових подій у консольному додатку на C#:

**Варіант 8,18**

Створіть консольну програму, яка симулює випадкові події. Це може бути симуляція погоди, вибір випадкових зображень, випадкових подій у грі тощо.

1. Користувач запускає програму.

2. Програма генерує випадкову подію на основі заздалегідь заданих параметрів. Наприклад, для симуляції погоди це можуть бути "сонячно", "хмарно", "дощ" та "сніг".

3. Згенерована подія виводиться на екран.

4. Користувач може запросити нову випадкову подію або завершити роботу програми.

При можливості структуруйте код з використанням функцій для повторного використання логіки.

Приклади додаткових можливих симуляцій:

– Симуляція випадкових температур залежно від часу доби.

- Симуляція випадкового вибору картки із колоди.

- Симуляція випадкових подій у текстовій грі (наприклад, пригоди персонажа).

**Варіант 9,19**

Створіть консольну програму, яка симулює кидок грального кубика. Користувач вводить кількість граней кубика і програма генерує випадковий результат, імітуючи кидок кубика з цією кількістю граней.

Користувач запускає програму.

Програма запитує у користувача кількість граней грального кубика (наприклад, 6, 10, 20 тощо).

Після отримання числа граней, програма генерує випадкове число від 1 до вказаної кількості граней і виводить результат на екран.

Користувач може запросити новий кидок знову, вибравши кількість граней, або завершити роботу програми.

**Варіант 10,20**

Створіть консольну програму, яка генерує випадкові дати та часи в заданих діапазонах. Користувач вводить початкову та кінцеву дати, а також інтервал часу, і програма генерує випадкові дати та часи у цьому діапазоні.

Користувач запускає програму.

Програма запитує у користувача початкову та кінцеву дати (включно) та інтервал часу (наприклад, 1 година, 30 хвилин).

Після отримання параметрів, програма генерує випадкову дату та час між зазначеними діапазонами із заданим інтервалом.

Згенеровані дати та часи виводяться на екран.

Користувач може запросити нові випадкові дати та часи знову, вказавши нові параметри, або завершити роботу програми.