|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TS\_lab7 |
| Назва проекта / ПЗ  Name of Project / Software | Ковальчук-task\_7\_1.exe |
| Рівень тестування  Level of Testing | Системний / System Testing |
| Автор тест-сьюта  Test Suite Author | Ковальчук Володимир Олегович |
| Виконавець  Implementer | Ковальчук Володимир Олегович |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування /  Test Result |
| TC-0 | 1.Зайдіть в код блокс ;  2.Задачу 7.2 перетворити в код  3.Введіть числа : 2 3 6 7 8 10 12 14 15 16 17 18 20 24 27 29 33 34 35 36 37 39 41 43 45 46 50 53 55 56 57 60 61 62 63 65 66 72 74 76 80 82 83 84 86 90 91 93 97 100 | 1.Виведення :2 100  3 97 ;6 93; 7 91; 8 90; 10 86; 12 84; 14 83; 15 82; 16 80;  17 76; 18 74;20 72;24 66;  27 65;29 63;33 62;34 61; 35 60;36 57;37 56;39 55;  41 53;43 50;45 46; | Passed |
| TC-1 | 1.Зайдіть в код блокс ;  2.Задачу 7.2 перетворити в код  3.Введіть числа :1 2 4 6 8 10 11 12 13 18 20 22 24 26 30 32 33 34 36 37 38 40 41 45 46 47 48 49 50 52 53 54 58 61 62 64 71 73 78 82 84 87 89 90 91 93 94 95 99 100 | 1.Виведення :1 100; 2 99; 4 95; 6 94; 8 93; 10 91; 11 90; 12 89; 13 87; 18 84; 20 82; 22 78; 24 73; 26 71; 30 64; 32 62; 33 61; 34 58; 36 54; 37 53; 38 52; 40 50; 41 49; 45 48; 46 47 | Passed |
| TC-2 | 1.Зайдіть в код блокс ;  2.Задачу 7.2 перетворити в код  3.Введіть числа : 21 25 27 39 54 63 66 73 77 80 82 92 105 106 121 123 124 130 133 143 148 151 178 185 186 189 196 209 218 234 264 277 278 284 286 294 297 301 307 320 322 328 335 344 346 350 361 368 378 379 | 1.Виведення : 21 379; 25 378; 27 368; 39 361; 54 350; 63 346; 66 344; 73 335; 77 328; 80 322; 82 320; 92 307; 105 301; 106 297; 121 294; 123 286; 124 284; 130 278; 133 277; 143 264; 148 234; 151 218; 178 209; 185 196; 186 189 | Passed |
| TC-3 | 1.Зайдіть в код блокс ;  2.Задачу 7.2 перетворити в код  3.Введіть числа : 3 5 7 13 16 25 26 28 38 44 49 59 64 65 68 73 78 82 84 85 87 89 92 98 114 117 118 121 125 126 130 131 134 135 145 146 157 161 162 163 164 170 171 175 178 181 182 187 196 198 | 1.Виведення : 3 198; 5 196; 7 187; 13 182; 16 181; 25 178; 26 175; 28 171; 38 170; 44 164; 49 163; 59 162; 64 161; 65 157; 68 146; 73 145; 78 135; 82 134; 84 131; 85 130; 87 126; 89 125; 92 121; 98 118; 114 117 | Passed |
| TC-4 | 1.Зайдіть в код блокс ;  2.Задачу 7.2 перетворити в код  3.Введіть числа : 3 9 19 20 21 22 35 36 44 54 57 61 62 84 86 90 99 100 103 111 112 115 129 131 132 134 137 141 144 146 150 155 156 158 160 161 162 163 167 170 174 176 179 181 183 184 185 189 192 197 | 1.Виведення : 3 197; 9 192; 19 189; 20 185; 21 184; 22 183; 35 181; 36 179; 44 176; 54 174; 57 170; 61 167; 62 163; 84 162; 86 161; 90 160; 99 158; 100 156; 103 155; 111 150; 112 146; 115 144; 129 141; 131 137; 132 134 | Passed |

|  |  |  |  |
| --- | --- | --- | --- |
| TC-5 | 1.Зайдіть в код блокс ;  2.Задачу 7.2 перетворити в код  3.Введіть числа : 2 5 9 11 12 25 30 33 37 38 46 48 49 50 55 58 59 60 62 73 74 77 78 83 85 86 90 99 101 102 103 104 105 107 109 110 116 118 122 124 125 127 130 136 139 180 187 188 193 199 | 1.Виведення :2 199; 5 193; 9 188; 11 187; 12 180; 25 139; 30 136; 33 130; 37 127; 38 125; 46 124; 48 122; 49 118; 50 116; 55 110; 58 109; 59 107; 60 105; 62 104; 73 103; 74 102; 77 101; 78 99; 83 90; 85 86 | Passed |
| TC-6 | 1.Зайдіть в код блокс ;  2.Задачу 7.2 перетворити в код  3.Введіть числа : 0 8 14 16 20 23 28 33 34 35 37 44 46 47 66 67 70 71 72 75 81 86 88 93 99 100 103 108 118 120 125 130 136 137 138 143 146 151 156 161 163 166 167 172 176 180 185 186 197 200 | 1.Виведення : 0 200; 8 197; 14 186; 16 185; 20 180; 23 176; 28 172; 33 167; 34 166; 35 163; 37 161; 44 156; 46 151; 47 146; 66 143; 67 138; 70 137; 71 136; 72 130; 75 125; 81 120; 86 118; 88 108; 93 103; 99 100 | Passed |
| TC-7 | 1.Зайдіть в код блокс ;  2.Задачу 7.2 перетворити в код  3.Введіть числа : 13 14 17 22 23 25 27 29 30 34 35 38 43 44 48 53 57 58 60 67 70 78 85 86 91 95 99 100 101 102 107 111 120 123 124 126 133 137 139 142 145 148 149 150 151 166 175 184 193 195 | 1.Виведення : 13 195; 14 193; 17 184; 22 175; 23 166; 25 151; 27 150; 29 149; 30 148; 34 145; 35 142; 38 139; 43 137; 44 133; 48 126; 53 124; 57 123; 58 120; 60 111; 67 107; 70 102; 78 101; 85 100; 86 99; 91 95 | Passed |
| TC-8 | 1.Зайдіть в код блокс ;  2.Задачу 7.2 перетворити в код  3.Введіть числа : 5 6 12 20 21 22 25 29 32 35 37 41 42 47 52 61 70 75 77 81 82 84 89 91 93 96 100 110 116 123 128 132 133 136 143 147 151 152 166 168 170 178 182 184 186 192 194 196 197 198 | 1.Виведення : 5 198; 6 197; 12 196; 20 194; 21 192; 22 186; 25 184; 29 182; 32 178; 35 170; 37 168; 41 166; 42 152; 47 151; 52 147; 61 143; 70 136; 75 133; 77 132; 81 128; 82 123; 84 116; 89 110; 91 100; 93 96 | Passed |
| TC-9 | 1.Зайдіть в код блокс ;  2.Задачу 7.2 перетворити в код  3.Введіть числа : 0 1 17 24 26 33 42 43 52 58 60 65 70 71 73 77 78 82 84 96 98 99 101 105 107 110 111 113 116 118 119 120 130 137 141 147 149 161 169 172 173 176 177 179 180 187 190 191 197 200 | 1.Виведення : 0 200; 1 197; 17 191; 24 190; 26 187; 33 180; 42 179; 43 177; 52 176; 58 173; 60 172; 65 169; 70 161; 71 149; 73 147; 77 141; 78 137; 82 130; 84 120; 96 119; 98 118; 99 116; 101 113; 105 111; 107 110 | Passed |

Висновки

1. Коректність алгоритму – алгоритм правильно реалізує поставлене завдання, працюючи згідно з описаною логікою.
2. Ефективність – використання масиву дозволяє отримати результат за O(n) часу.
3. Чіткість вихідних даних – вивід відповідає вимогам: пари розділені "; ".
4. Простота реалізації – використано базові конструкції C (масив, цикл for, printf).
5. Мінімальне споживання пам’яті – використовується лише один масив без додаткових структур
6. Гнучкість – програму можна легко адаптувати для роботи з іншою кількістю чисел.
7. Універсальність – код працює на будь-яких платформах, де доступний компілятор C.
8. Відсутність зайвих обчислень – кожен елемент використовується лише один раз.
9. Відсутність дублювання даних – пари формуються динамічно без створення додаткових масивів.
10. Масштабованість – алгоритм легко розширити для масивів більшого розміру.
11. Коректна нумерація індексів – програма не виходить за межі масиву.
12. Відповідність формату виводу – кожна пара відділена "; ", що відповідає вимогам.
13. Обробка всіх елементів – жоден елемент не втрачається.
14. Використання циклу з правильним кроком – ітерація триває 25 разів, щоб не дублювати пари.
15. Перевірка правильності виконання – тестування на різних наборах вхідних даних
16. Зрозумілість коду – проста структура, зрозуміла навіть новачкам.
17. Легкість редагування – легко змінити формат виводу або додати функціонал.
18. Читабельність – кожен крок логічно структурований.
19. Використання коментарів – у коді є пояснення основних етапів.
20. Структурованість – логічно розділені етапи програми.
21. Робота з від’ємними числами – програма коректно працює з будь-якими цілими числами.
22. Робота з нульовими значеннями – не викликає помилок.
23. Обробка однакових чисел – програма не змінює їх порядок.
24. Обробка випадкових чисел – результат залишається правильним.
25. Перевірка на межах – працює коректно з найбільшими та найменшими значеннями типу int.
26. Користувачеві легко ввести дані – використовується стандартний scanf.
27. Формат виводу простий для читання – чітко розділені пари.
28. Можливість змінити формат виводу – легко адаптується під інші вимоги.
29. Швидке виконання – програма виконується за мілісекунди.
30. Не потребує додаткових бібліотек – працює з stdio.h.
31. Можливість перевірки введених даних – можна додати валідацію.
32. Мінімальна ймовірність переповнення – масив має фіксований розмір.
33. Коректне використання індексів – немає виходу за межі масиву.
34. Робота з будь-якими числами – підтримує будь-які int.
35. Сумісність з різними компіляторами .
36. Можна розширити до роботи з файлами – легко зчитувати дані з файлу.
37. Можна зробити функціональним – винести логіку у функцію.
38. Можливість реалізувати рекурсивно – альтернативний підхід.
39. Оптимізація під багатопоточність – можна розділити обчислення.
40. Додавання сортування – можна спочатку впорядкувати дані.
41. Легко адаптувати до інших мов – можна реалізувати на Python, Java.
42. Можна розширити для інших структур даних – наприклад, списки.
43. Можна зробити графічний інтерфейс – інтерактивний ввід даних.
44. Можливість використання в реальних завданнях – аналіз послідовностей.
45. Додати більше форматів виводу.
46. Корисний досвід – практика роботи з масивами та циклами
47. Закріплення знань C – особливо щодо введення та виведення.
48. Покращення логічного мислення – аналіз структури послідовності.
49. Розуміння важливості правильного форматування – як зробити вивід зручним.
50. Мотивує до подальшого вивчення C – після цієї задачі хочеться розвиватися далі.