

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Н.Д. Любашенко, І.А. Костюшко

**Програмування.
Частина 2.
Вказівки до виконання
лабораторних робіт**

*Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для студентів,
які навчаються за спеціальністю 113 «Прикладна математика»*

Київ
КПІ ім. Ігоря Сікорського
2023

Програмування. Частина 2. Вказівки до виконання лабораторних робіт [Електронний ресурс]: навч. посіб. для студ. спеціальності 113 «Прикладна математика», спеціалізації «Наука про дані та математичне моделювання» / КПІ ім. Ігоря Сікорського ; уклад.: Н.Д. Любашенко, І. А. Костюшко. – Електронні текстові дані (1 файл: 28 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2023. – 75 с.

*Гриф надано Методичною радою КПІ ім. Ігоря Сікорського
(протокол № 8 від 02.06.2023 р.)
за поданням Вченої ради факультету прикладної математики
(протокол № 10 від 20.05.2023 р.)*

Електронне мережне навчальне видання

Програмування. Частина 2. Вказівки до виконання лабораторних робіт

Укладачі: *Любашенко Наталія Дмитрівна, ст. викл.
Костюшко Ірина Анатоліївна, канд. фіз.-мат. наук, доц.*

Відповідальний редактор: *Бай Ю.П., канд. фіз.-мат. наук, ст. викл.*

Рецензенти: *Пакриш О.Є., канд. техн. наук, доц.*

Посібник адресований студентам закладів вищої освіти зі спеціальності «Прикладна математика», спеціалізації «Наука про дані та математичне моделювання». Матеріал відповідає змісту міжнародного стандарту ISO/IEC з мови C. Цикл робіт охоплює всі базові теми, важливі для практичного застосування мови C: середовище програмування, введення та виведення, концепція типів, функції, макроси, бібліотека C, управління пам'яттю. Посібник складається з восьми розділів. Кожний розділ надає коротку інформацію по темі лабораторної роботи, приклади програм, індивідуальні завдання, контрольні питання для самоперевірки.

Викладений матеріал відповідає діючим стандартам та навчальним програмам з дисципліни «Програмування».

Зміст

ВСТУП	5
Лабораторна робота № 1. С – мова процедурного програмування.....	6
Коротка довідкова інформація	6
Завдання для всіх варіантів	8
КОНТРОЛЬНІ ПИТАННЯ.....	8
Лабораторна робота № 2. Концепція типів. Стандартне введення та виведення.....	10
Коротка довідкова інформація	10
Завдання для всіх варіантів	12
КОНТРОЛЬНІ ПИТАННЯ.....	13
Лабораторна робота № 3. Похідний тип масив. Робота з файлами.....	14
Коротка довідкова інформація	14
Завдання для всіх варіантів	18
КОНТРОЛЬНІ ПИТАННЯ.....	18
Лабораторна робота № 4. Масив типу char та масив типу вказівник на char.....	19
Коротка довідкова інформація	19
Завдання для всіх варіантів	20
КОНТРОЛЬНІ ПИТАННЯ.....	21
Лабораторна робота № 5. Похідний тип структура.....	22
Коротка довідкова інформація	22
Завдання для всіх варіантів	24
КОНТРОЛЬНІ ПИТАННЯ.....	24
Лабораторна робота № 6. Робота з оперативною пам'яттю	25
Коротка довідкова інформація	25
Завдання для всіх варіантів	26
КОНТРОЛЬНІ ПИТАННЯ.....	27
Лабораторна робота № 7. Функції та макроси	28
Коротка довідкова інформація	28
Завдання для всіх варіантів	29
КОНТРОЛЬНІ ПИТАННЯ.....	29
Лабораторна робота № 8. Бібліотечні функції	30
Коротка довідкова інформація	30
Завдання для всіх варіантів	31
КОНТРОЛЬНІ ПИТАННЯ.....	32
Додаток А. Елементи мови С згідно стандарту	33
Додаток Б. Індивідуальні завдання до ЛР № 1	34

	4
Додаток В. Таблиці формул фізики.....	36
Додаток Г. Індивідуальні завдання до ЛР № 2.....	43
Додаток Д. Індивідуальні завдання до ЛР № 3	45
Додаток Е. Індивідуальні завдання до ЛР № 4.....	53
Додаток Ж. Індивідуальні завдання до ЛР № 5.....	54
Додаток К. Індивідуальні завдання до ЛР № 6	69
Додаток Л. Індивідуальні завдання до ЛР № 7	72
Додаток М. Індивідуальні завдання до ЛР № 8.....	73
ЛІТЕРАТУРА	75

ВСТУП

Мова програмування С має поважний вік, але її актуальність не втрачена. Мова успішно використовується в розробці ефективних системних та прикладних програм. Її вивчення дозволяє спеціалісту глибше ознайомитися з принципами роботи обчислювальних систем, з організацією обробки даних всередині комп'ютера. Такі знання суттєво допоможуть будь-якому програмісту у вивченні інших мов вищого рівня. Звичайно, далеко не всі програмісти володіють цією мовою, але знають про її внесок у розвиток сучасного програмного забезпечення.

Мова С не є легкою для вивчення, проте пропонує програмісту широкі можливості, гнучкий підхід до до реалізації багатьох-багатьох задач.

Дисципліна “Програмування” передбачає ознайомлення з мовою програмування С, а в межах вивчення мови виконання циклу лабораторних робіт.

Даний навчальний посібник призначено для проведення лабораторних робіт з кредитного модуля “Програмування. Частина 2”.

Метою виконання восьми лабораторних робіт є формування у студентів практичних навичок з використання парадигми імперативного програмування. Виконання цих робіт дозволяє студентам зрозуміти елементи мови, концепції мови, призначення програмної С-бібліотеки.

Завдання підбирались таким чином, щоб студенти відчували гнучкість мови, можливість програмування того чи іншого алгоритму різними способами.

Дуже важливим є те, щоб під час виконання цих робіт студенти опрацювали питання розробки ефективних за часом та ресурсами пам'яті програм.

Однією з основних концепцій мови С є типізація даних, тому різні типи розглядаються практично в усіх роботах.

Кожній лабораторній роботі присвячено окремий розділ посібника. У розділах наводяться короткі теоретичні відомості за темою роботи, приклади програм для допомоги студентам у виконанні індивідуального завдання, формулюється загальне завдання для всіх варіантів.

Індивідуальні завдання оформлено в табличному вигляді та подано у додатку.

При цьому дані наводяться у “сирому” вигляді, а студенту потрібно ці дані структурувати, обрати для них відповідні типи мови С і тільки після цього починати писати програму згідно варіанту.

Необхідні довідкові матеріали також розташовані у додатку.

Матеріал посібника відповідає силабусу з кредитного модуля “Програмування. Частина 2”.

Лабораторна робота № 1. С – мова процедурного програмування

Мета роботи: отримати початкові знання з мови програмування С, навчитись працювати в інтегрованому середовищі програмування (IDE).

Коротка довідкова інформація

Мова С реалізує парадигму процедурного програмування.

Ця парадигма полягає в тому, що програмний код поділяється на окремі блоки (процедури), які виконують ту чи іншу задачу. Процедури (інші назви - підпрограми, функції) містять серію обчислювальних кроків – команд, які виконуються в порядку, визначеному програмістом, тобто команда за командою. Якщо команди поміняти місцями, порядок виконання програми також зміниться.

Виклик процедури означає її запуск з передачею аргументів.

Після виконання процедура може повертати в місце виклику значення, її виконання закінчується. Далі виконується наступна за викликом процедури команда викликаючої програми.

В мові С процедура називається функцією.

В довідковій таблиці (табл. А.1), що відповідає стандарту [2], наведено всі конструкції мови С – граматичні елементи, з яких будується текст С-програми для реалізації того чи іншого алгоритму обробки тих чи інших даних.

Граматичними елементами є:

- директиви препроцесора,
- послідовність тверджень та блоків, які відтворюють алгоритм,
- вирази у твердженнях,
- декларації та зовнішні визначення (є обов'язковими для всіх даних та функцій),
- лексеми, тобто елементарні мовні одиниці, з яких будуються всі перелічені вище елементи.

Граматика мови С не є простою, але її вивчення дозволить писати гнучкі, ефективні, переносимі програми.

На рис. 1.1 подано текст відомої програми, яка виводить на екран ялинку. Користувач задає символ, яким малюється ялинка, і також ширину та висоту.

Дана програма, як і будь-яка інша програма на мові С, складається з граматичних елементів. Їх повний перелік наведено в табл. А.1. Що стосується алгоритму, реалізованого програмою, пропонується розібратися в ньому самостійно, що є частиною підготовки до виконання даної лабораторної роботи.

В даній роботі передбачається використання циклів, які в мові C називаються твердженнями ітерації. Зокрема, твердження `for` використовується для повторення певного блоку коду (операторів) задану кількість разів.

Твердження ітерації `for` (рис. 1.2) має синтаксис `for (речення-1; вираз-2; вираз-3)` і працює наступним чином. Вираз-2 є управляючим виразом, який обчислюється перед кожним виконанням тіла циклу. Вираз-3 обчислюється після кожного виконання тіла циклу. Якщо речення-1 є декларацією, то межа видимості відповідних ідентифікаторів – це весь цикл, включаючи також інші два вирази; реалізується в порядку виконання перед першим обчисленням управляючого виразу. Якщо речення-1 є виразом, він обчислюється до першого обчислення управляючого виразу.

Примітка: `for` повинен мати дві крапки з комою між відкриваючою та закриваючою дужками.

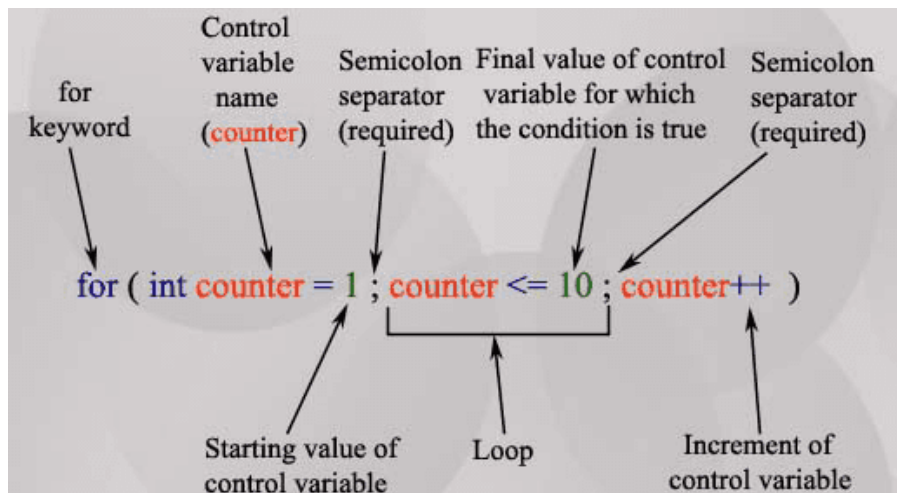


Рисунок 1.2 – Твердження `for`

Завдання для всіх варіантів

1. Обрати зручне середовище для виконання циклу лабораторних робіт з мови C, наприклад, Visual Studio Code, Code::Blocks, Replit, GDB Online, NetBeans.
2. Переконавшись, що програма, яка малює на екрані стилізовану ялинку (рис. 1.1), працює в обраному середовищі. Ознайомитися з текстом програми, знайти в ньому граматичні елементи мови C, зрозуміти їх смисл.

Індивідуальні завдання

1. Виконати індивідуальне завдання (табл. Б.1), внісши у дану програму відповідні зміни; додати коментарі до оновлених рядків коду.
2. Оформити стислий звіт з роботи у форматі pdf-документа, який складається з таких частин: титул, постановка задачі, тексти початкової та зміненої програм, скріншоти результатів виконання обох програм.

КОНТРОЛЬНІ ПИТАННЯ

1. Поясніть парадигму процедурного програмування.
2. Що таке інтегроване середовище розробки (IDE)?

3. Чому розробка стандартів мов програмування є важливою для програмістів?
4. Назвіть переваги та недоліки мови програмування С.
5. З яких частин складається типова С-програма?

Лабораторна робота № 2. Концепція типів. Стандартне введення та виведення

Мета роботи: ознайомлення з базовими типами даних та їх застосуванням для проведення обчислень математичних моделей, основними стандартними функціями введення та виведення даних.

Коротка довідкова інформація

Типізація даних є однією з концепцій мови C. Вона означає, що програміст повинен явно або неявно описати типи всіх об'єктів програми, а також типи значень, які повертаються функціями. Від типу залежатиме спосіб та форма зберігання та обробки даних системою. Тип задається програмістом в декларації об'єкта або функції.

Конкретний тип можна описати, використовуючи так звані базові та/або похідні типи. Зокрема, всі прийняті в C типи наведено в табл 2.1.

Таблиця 2.1 Класифікація типів

ТИП ПЕРЕЛІКУ	БАЗОВІ ТИПИ					ПОХІДНІ ТИПИ	
	enum	Стандартні цілі типи		Типи з плаваючою точкою		Масив	
		Стандартні цілі знакові типи	Стандартні цілі беззнакові типи	Дійсні типи	Комплексні типи		Структура
		signed char short int int long int long long int	unsigned char unsigned short int unsigned int unsigned long int unsigned long long int _Bool				
	char	Розширені цілі знакові типи, визначені реалізацією	Розширені цілі беззнакові типи, визначені реалізацією			Функція	
						Вказівник	
						Атомарний	
Тип void охоплює порожній набір значень							

Введення та виведення даних реалізовано у мові C не спеціальними операторами, як у деяких інших мовах, а функціями. Ці функції містяться у відповідних стандартних бібліотеках, зокрема, <stdio.h>. В табл. 2.2 наведено прототипи функцій scanf та printf для форматowanego введення (з клавіатури) та виведення (на екран).

Таблиця 2.2

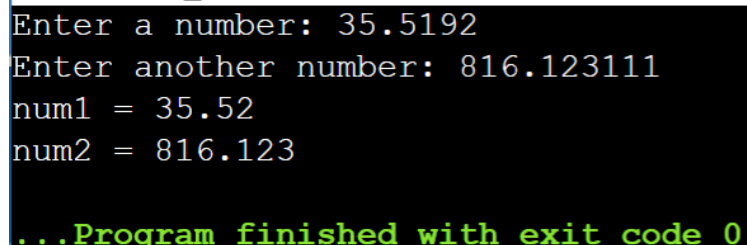
Декларація функції	Призначення функції
<pre>int printf(const char * restrict format, ...);</pre> повертає кількість переданих символів або від'ємне значення при виникненні помилки.	Виведення даних відбувається під управлінням рядка формату. Формат визначає, як перетворюватимуться наступні аргументи перед виведенням.
<pre>int scanf(const char * restrict format, ...);</pre> повертає значення макросу EOF, якщо вхідна помилка відбувається до завершення першого перетворення (якщо таке є). В іншому випадку функція повертає кількість введених елементів.	Введення відбувається під управлінням рядка формату. Формат визначає, як перетворюватимуться введені дані при присвоєнні наступним аргументам. Ці аргументи є вказівниками на об'єкти, в які попадуть перетворені дані.

В наступній програмі (рис. 2.1) використано функцію `printf` для виведення на екран запрошення користувачеві ввести з клавіатури число (рядки коду 7, 9). Введення здійснюється за допомогою стандартної функції `scanf` (рядки коду 8, 10). В рядках 12, 13 коду запрограмовано форматоване виведення введених чисел на екран.

```

1  #include <stdio.h>
2  int main()
3  {
4      float num1;
5      double num2;
6
7      printf("Enter a number: ");
8      scanf("%f", &num1);
9      printf("Enter another number: ");
10     scanf("%lf", &num2);
11
12     printf("num1 = %0.2f\n", num1);
13     printf("num2 = %0.3lf", num2);
14
15     return 0;
16 }
17

```



```

Enter a number: 35.5192
Enter another number: 816.123111
num1 = 35.52
num2 = 816.123
...Program finished with exit code 0

```

Рис. 2.1 – Форматоване введення та виведення

Мова С є гарним вибором для проведення обчислень. На рис. 2.2 дано приклад програми для обчислення моделі руху.

```

main.c
1  /*****
2  Програма обчислення швидкості рівномірного руху
3  S - шлях
4  t - час
5  *****/
6  #include <stdio.h>
7  float Speed()
8  { float S, t;
9      printf(" S (в км) = ");
10     scanf("%f",&S);
11     printf(" t (в год.)= ");
12     scanf("%f",&t);
13
14     // формула фізики
15     return (S/t);
16 }
17 int main()
18 {
19
20     // результат обчислень
21     printf("speed v = %0.3f", Speed());
22
23     return 0;
24 }
25
S (в км) = 134.5
t (в год.)= 2.5
speed v = 53.800

...Program finished with exit code 0
Press ENTER to exit console.

```

Рисунок 2.2 – Приклад програми обчислення швидкості рівномірного руху

Завдання для всіх варіантів

1. Знайти за номером варіанту в додатку Г індивідуальне завдання, потім в додатку В – формули. Ознайомитись з ними.
2. Написати програму для проведення обчислень за формулами з фізики. Попередньо обрати тип для кожного з операндів та результату обчислень. Для отримання коректного результату важливо звернути увагу на одиниці вимірювання фізичних величин.
3. Програма має виконувати такі функції:
 - 1) надати користувачеві можливість обирати формулу;
 - 2) запросити користувача ввести вихідні дані з клавіатури для обраної формули;
 - 3) провести обчислення за обраною формулою;
 - 4) вивести результат на екран у зручній для користувача формі;
 - 5) зробити скріншоти виконання контрольних прикладів, додати в звіт.

Індивідуальні завдання

1. Для кожного варіанта (табл. Г.1) надано номери таблиць (табл. В.1–В.7) та номери формул в таблиці, які треба програмно реалізувати.

КОНТРОЛЬНІ ПИТАННЯ

1. Поясніть концепцію типів даних в мові С.
2. Наведіть основну класифікацію типів в С.
3. Що таке тип, визначений реалізацією?
4. Як реалізовано введення та виведення в С?
5. Які функції обміну даними є в С для роботи з клавіатурою та монітором?

Лабораторна робота № 3. Похідний тип масив. Робота з файлами

Мета роботи: ознайомлення з похідним типом масив та організацією обміну даними в середовищі С-програмування.

Коротка довідкова інформація

Згідно стандарту, тип масив описує непорожню множину об'єктів, яка розміщується в неперервному шматку пам'яті, з певним типом кожного об'єкта, який називається типом елемента.

Тип масив характеризується типом елементів і кількістю елементів в масиві. Тип є похідним тому, що походить з типу його елементів, і якщо тип елемента – `int`, тип масив часто називають "масивом `int`".

Якщо вираз типу масив зустрічається в будь-якому контексті (окрім окремих випадків, наприклад, операндів у `sizeof`), то він піддається компілятором неявному перетворенню у вказівник на його перший елемент.

Приклади декларацій типів масив:

```
int ms1[19];
```

```
int ms2[] = { 21, 34, 59 };
```

```
int matrix[4][3] = {{ 11, 22, 33 }, { 44, 55, 66 }, { 77, 88, 99 }, { 100, 101, 102 }};
```

На рис. 3.1, 3.2 показано, що до елементів масиву можна досягти як за допомогою оператора індексації `[]`, так і за допомогою вказівників.

```

main.c
1  /******
2  Масиви та вказівники
3  *****/
4  #include <stdio.h>
5  int main()
6  {
7      int msv[3];
8      msv[0]=11;
9      msv[1]=22;
10     msv[2]=33;
11     printf("\n Елемент = %d ", msv[1]);
12     printf("\n Елемент = %d ", *(msv+1));
13     return 0;
14 }
15
Елемент = 22
Елемент = 22
  
```

Рисунок 3.1 – Оператор індексації `[]`

```

main.c
1  /* масиви та вказівники */
2  #include <stdio.h>
3  int main()
4  {
5      int msv[3];
6      *msv=11;
7      *(msv+1)=22;
8      *(msv+1)=33;
9      printf("\n Елемент = %d ", *(msv+1));
10     return 0;
11 }
12
✓ ↗ ↵
Елемент = 33
...Program finished with exit code 0

```

Рисунок 3.2 – Оператор * (вміст за адресою)

Для виконання роботи можна застосувати стандартні функції для роботи з файлами – функції для форматowanego та символного введення-виведення (табл. 3.1) .

Таблиця 3.1 – Функції для форматowanego та символного ведення-виведення

Декларація функції	Призначення функції
<code>int fscanf(FILE * restrict stream,const char * restrict format, ...);</code> повертає значення макросу EOF , якщо вхідна помилка відбувається до завершення першого перетворення (якщо таке є). В іншому випадку функція повертає кількість введених елементів	Читає введення з потоку, на який вказує перший аргумент. Введення відбувається під управлінням рядка - формату, на який вказує другий аргумент. Формат визначає, як перетворюватимуться введені дані при присвоєнні наступним аргументам. Ці аргументи є вказівниками на об'єкти, в які попадуть перетворені дані. Якщо аргументів для формату недостатньо, поведінка програми не визначена.
<code>int fprintf(FILE * restrict stream,const char * restrict format, ...);</code> повертає кількість переданих символів або від'ємне значення при виникненні помилки	Записує виведення у потік, на який вказує перший аргумент. Виведення даних відбувається під управлінням рядка - формату, на який вказує другий аргумент. Формат визначає, як перетворюватимуться наступні аргументи перед виведенням. Якщо аргументів для формату недостатньо, поведінка не визначена.
<code>int fgetc(FILE *stream);</code> повертає наступний символ з потоку введення або EOF, якщо трапилась помилка або досягнуто кінець файлу.	Отримує символ як тип <code>unsigned char</code> , перетворює у тип <code>int</code> , просуває індикатор позиції у файлі для вказаного в аргументі потоку.

<pre>char *fgets(char * restrict s, int n, FILE * restrict stream);</pre> повертає вміст s або null-вказівник при помилці читання.	Читає з вказаного потоку (третій параметр) в масив, на який вказує перший параметр. Кількість прочитаних символів, включаючи символ нового рядка, не перевищує значення другого аргумента. Null -символ записується відразу після того, як в масив прочитано останній символ.
<pre>int fputc(int c, FILE *stream);</pre> повертає записаний символ. Якщо виникає помилка запису, повертається EOF.	Пише символ, заданий першим аргументом (перетворений в unsigned char), до заданого потоку, в позицію, яка відповідає значенню індикатора.
<pre>int fputs(const char * restrict s, FILE * restrict stream);</pre> повертає EOF, якщо виникає помилка запису; в іншому випадку повертає невід’ємне значення.	Записує рядок, на який вказує перший аргумент, до вказаного потоку (другий аргумент). Завершаючий Null -символ не записується.

Приклад. Програма (рис. 3.3) призначена для виведення вмісту заданого користувачем файлу на екран. Символьні дані зберігаються у деякому файлі (рис. 3.4). При виконанні програми ім’я файлу вводять користувач (рядки 18, 19). Після цього програма відкриває заданий файл (рядок 21), читає його вміст у масив line (рядки 23-27), а потім виводить на екран. Розміри масиву задаються директивою #define препроцесора (рядки 7-8). Звичайно, можна декларувати цей тип іншими способами, наприклад, line[100][100].

The image shows a code editor with a file named `main.c` and a file named `kilt`. The code in `main.c` is a C program that reads a file named `kilt` and prints its contents. The program uses `stdio.h`, `stdlib.h`, and `string.h`. It defines constants `C` (100), `R` (100), and `FNAME` (15). The `main` function opens the file `kilt` in read mode, reads lines into an array `line`, and prints them. The output of the program is shown in a terminal window below the code editor.

```

1  /* Програма читає файл у двовимірний символний масив */
2
3  #include <stdio.h>
4  #include <stdlib.h>
5  #include <string.h>
6
7  #define C 100
8  #define R 100
9  #define FNAME 15
10 int main(void)
11 {
12     char line[R][C];
13     char fname[FNAME];
14     FILE *fptr = NULL;
15     int i = 0;
16     int nur = 0;
17
18     printf(" File name to be opened : ");
19     scanf("%s",fname);
20
21     fptr = fopen(fname, "r");
22
23     while(fgets(line[i], C, fptr))
24     {
25         line[i][strlen(line[i]) - 1] = '\0';
26         i++;
27     }
28     nur = i;
29     printf("\n The content of the file %s is : \n",fname);
30     for(i = 0; i < nur; ++i)
31     {
32         printf(" %s\n", line[i]);
33     }
34     printf("\n");
35     return 0;
36 }

```

File name to be opened : kilt

The content of the file kilt is :

Dina Fogel
Mark Gerdty
Lisa van Berdt

...Program finished with exit code 0

Рисунок 3.3 – Програма та результати виведення

main.c	kilt
1	Dina Fogel
2	Mark Gerdty
3	Lisa van Berdt
4	

Рисунок 3.4 – Вміст файлу kilt

Завдання для всіх варіантів

1. Створити текстовий файл та вручну заповнити його даними з варіанту.
2. Написати програму для виконання індивідуального завдання; використати двовимірний символьний масив фіксованого розміру для зберігання даних в пам'яті; обробку елементів робити за допомогою оператора індексації [].

Індивідуальні завдання за варіантом:

1. Для кожного варіанта надано номери таблиць (табл. В.1–В.7) та номери формул в таблиці (табл. Г.1), які треба програмно реалізувати.

КОНТРОЛЬНІ ПИТАННЯ

1. Що містить декларація типу масив в мові C?
2. Як інтерпретується оператор індексації [] в мові C?
3. Які бібліотечні функції зручно використовувати для файлового введення та виведення даних типу char?
4. Чому тип масив є похідним типом?

Лабораторна робота № 4. Масив типу char та масив типу вказівник на char

Мета роботи: поглиблене освоєння типів масив та вказівник.

Коротка довідкова інформація

В лабораторній роботі №3 розглядався тип масив масивів (двовимірний масив). Цей масив в програмі заповнювався даними – рядками символів різної довжини, тому деякі елементи масиву (і відповідна пам'ять) не використовувались.

З метою економії пам'яті комп'ютера можна використати тип вказівник.

Згідно стандарту, тип вказівника походить з типу функції або з типу об'єкта, на які може посилатись цей вказівник. Оскільки в програмуванні посилення означає адресування до об'єкта, тобто звертання за конкретною адресою в оперативній пам'яті, стає зрозумілим, що робота програміста з типом вказівник є роботою з адресами комірок пам'яті.

Для скорочення назви типу в декларації, наприклад, `int *ptr`, замість фрази “ptr має тип вказівник, що походить від типу `int`, на який можна посилатись”, кажуть коротко: ptr має тип "вказівник на `int`".

Отже, значеннями об'єкта `ptr` можуть бути адреси комірок пам'яті, в яких містяться значення типу `int`.

Над даними типу вказівник можна проводити дії адресної арифметики. Додавання або віднімання числа `n` до вказівника означає просування по комірках пам'яті, а саме збільшення або зменшення адреси на величину `n*sizeof(тип-вказівника)` байтів. Наприклад, нехай `ptr` містить адресу 1004. Також нехай тип `int` займає чотири байти. Тоді вираз `ptr++` матиме значення 1008.

Нехай в програмі потрібно розмістити символьні дані в масиві для подальшої обробки. Розглянемо, як можна це робити.

Перший спосіб. Декларується та ініціалізується масив `name` фіксованого розміру:
`char name[3][20] = { "Christopher", "Kevin", "Roy"};`

Система виділить пам'ять для 60-ти елементів (`3*20`) типу `char` масиву `name`, також початкові значення 22-х символів заносяться у відповідні елементи масиву.

Другий спосіб. Декларується масив `name` вказівників з 3-х елементів, кожний з яких ініціалізується адресою відповідного символьного рядка-літерала:

`char *name[3] = { "Christopher", "Kevin", "Roy"};`

Другий спосіб дає економію пам'яті за рахунок того, що літерали займають пам'ять, яка точно відповідає їх довжині. Додатково маємо пам'ять для 3-х елементів - адрес. Нехай кожний елемент масиву вказівників займає 8 байтів. Тоді загальний об'єм потрібної пам'яті буде 46 (`22+3*8`) байтів, що є менше, ніж у першому способі.

Програма (рис. 4.1) обчислює об'єм пам'яті, виділеної системою для зберігання 12-ти рядків у масиві `char Stars[12][20]`.

```

main.c
1
2 #include <stdio.h>
3 int main(int argc, char * argv[])
4 {
5     char Stars[12][20]={"Aquarius", "Libra", "Gemini", "Capricorn",
6                         "Virgo", "Taurus", "Sagittarius", "Leo",
7                         "Aries", "Scorpio", "Cancer", "Pisces"};
8
9     printf(" memory size = %ld", sizeof(Stars));
10 }
11
memory size = 240
...Program finished with exit code 0

```

Рисунок 4.1 – Пам'ять під масив символічних рядків

Друга програма (рис. 4.2) обчислює об'єм пам'яті для зберігання 12-ти рядків за допомогою масива вказівників з 12-ти елементів.

```

main.c
1 #include <stdio.h>
2 int main(int argc, char * argv[])
3 {
4     char *Stars[]={"Aquarius", "Libra", "Gemini", "Capricorn",
5                   "Virgo", "Taurus", "Sagittarius", "Leo",
6                   "Aries", "Scorpio", "Cancer", "Pisces"};
7
8     printf(" memory size = %ld",
9           sizeof(Stars)+
10          sizeof("Aquarius")+sizeof("Libra")+
11          sizeof("Gemini")+sizeof("Capricorn")+
12          sizeof("Virgo")+sizeof("Taurus")+sizeof("Sagittarius")+
13          sizeof("Leo")+sizeof("Aries")+sizeof("Scorpio")+
14          sizeof("Cancer")+sizeof("Pisces"));
15 }
16
input
memory size = 185
...Program finished with exit code 0

```

Рисунок 4.2 – Застосування вказівників для економії пам'яті

Результати: 240 байтів для першої програми, 185 байтів для другої програми.

Завдання для всіх варіантів

1. Декларувати та ініціалізувати масив вказівників адресами рядків-літералів, взятих з індивідуального завдання.
2. Обробку символів зробити за допомогою оператора * (вміст за адресою).
3. Запрограмувати обробку даних, як вказано у варіанті індивідуального завдання.

Індивідуальні завдання

1. Індивідуальні завдання містяться в таблиці (табл. Е.1).

КОНТРОЛЬНІ ПИТАННЯ

1. Як пов'язані типи масив та вказівник в мові С?
2. Які оператори використовують для роботи з вказівниками?
3. Що таке рядок-літерал?
4. Чому тип вказівник є похідним типом?

Лабораторна робота № 5. Похідний тип структура

Мета роботи: ознайомлення з похідним типом структура.

Коротка довідкова інформація

Структура – це агрегований похідний тип даних, який дозволяє об'єднувати елементи даних різних типів.

Наприклад, в програмі потрібно обробляти дані про книги. Нехай ці дані структуровані так: для кожної книги задано автора, назву, анотацію та номер. Тоді доцільно обрати для зберігання цих даних наступний структурний тип мови C, декларація якого записується наприклад, так:

```
struct Books { char title[50];
char author[50];
char subject[100];
int book_id; } book;
```

Ця декларація містить ідентифікатор об'єкта book. Його тип – структура з тегом Books, яка складається з 4-х елементів (title, author, subject, book_id) різних типів.

Щоб отримати доступ до елементів структури, можна скористатись операторами:

- оператор елемента `.` ;
- оператор вказівника на структуру `->` .

Приклад. У файлі library.txt зберігаються структуровані дані про книги (рис. 5.1). Програма (рис. 5.2) відкриває файл користувача з іменем library.txt в режимі читання (рядки 14-15). Далі за допомогою тверджень ітерації та функції fscanf файл читається в масив з 3-х елементів структурного типу, і далі дані виводяться на екран (рис. 5.2). Результат роботи програми наведено на рис. 5.3

main.c	library.txt	:
1	Dracula*Bram Stoker*Vampires*1991	
2	Polina*Henry James*Love*1884	
3	Philosopher's Stone*Joanne Rowling*fiction*1997	
4		

Рисунок 5.1 – Файл структурованих даних

main.c	library.txt	:
1	#include<stdio.h>	
2	#include <stdio.h>	
3	struct Books {	
4	char title[50];	
5	char author[50];	
6	char subject[100];	
7	int book_id;	
8	};	
9		
10	int main()	
11	{	
12	struct Books b[3];	
13	int i,j;	
14	FILE *p;	
15	p = fopen("library.txt", "r");	
16	for (j=0;j<3;j++)	
17	{ i=-1;	
18	do	
19	{ i=i+1;	
20	//printf("i=%d",i);	
21	fscanf(p,"%c",&b[j].title[i]);	
22	printf("%c",b[j].title[i]);	
23	}	
24	while (b[j].title[i]!='*');	
25	i=-1;	
26	do	
27	{ i=i+1;	
28	fscanf(p,"%c",&b[j].author[i]);	
29	printf("%c",b[j].author[i]);	
30	}	
31	while (b[j].author[i]!='*');	
32	i=-1;	
33	do	
34	{ i=i+1;	
35	fscanf(p,"%c",&b[j].subject[i]);	
36	printf("%c",b[j].subject[i]);	
37	}	
38	while (b[j].subject[i]!='*');	
39	i=-1;	
40	i=i+1;	
41	fscanf(p,"%d",&b[j].book_id);	
42	printf("%d",b[j].book_id);	
43	}	
44	return 0;	
45	}	

Рисунок 5.2 – Програма обробки даних структурного типу

```

1  #include<stdio.h>
2  #include <stdio.h>
3  struct Books {
4      char title[50];
5      char author[50];
6      char subject[100];
7      int book_id;
8  };
9
10 int main()
11 {

```

Dracula*Bram Stoker*Vampires*1991
 Polina*Henry James*Love*1884
 Philosopher's Stone*Joanne Rowling*fiction*1997
 ...Program finished with exit code 0

Рисунок 5.3 – Результати виведення елементів структури

Завдання для всіх варіантів

1. В кожному варіанті завдання є таблиця з даними з різних предметних областей. Для роботи з табличними даними в С-програмі зручно скористатись типом масив з елементів структурного типу. Тоді кожний рядок таблиці можна записати у відповідний елемент масиву.
2. Розробити структурний тип для таблиці з варіанту. Записати його на початку програми.
3. Внести табличні дані вручну у текстовий файл (з довільною назвою) у відповідності з розробленою структурою.
4. Написати програму, яка б вводила дані в масив, елементи якого мають тип структура з п. 2, та обробляла їх за варіантом завдання.

Індивідуальне завдання

1. Індивідуальні завдання містяться в таблиці (табл. Ж.1).

КОНТРОЛЬНІ ПИТАННЯ

1. Які твердження ітерації є в мові С?
2. Які оператори дозволяють звернутися до елемента структури в мові С?
3. Що таке тег структури і як він використовується в С програмі?
4. Чому тип структура є похідним типом?

Лабораторна робота № 6. Робота з оперативною пам'яттю

Мета роботи: ознайомлення зі стандартними функціями управління пам'яттю.

Коротка довідкова інформація

Мова С відома тим, що надає програмісту можливість управляти оперативною пам'яттю, а саме, надати запит до системи (через виклики відповідних функцій) на виділення пам'яті та отримати адресу початку виділеного шматка.

Для динамічного розподілу пам'яті використовуються бібліотечні функції `malloc()`, `calloc()`, `realloc()` і `free()`. Ці функції визначено у файлі заголовка `<stdlib.h>`.

Синтаксис функції `malloc`: `void *malloc(size_t size)`, де параметр `size` типу `size_t` означає розмір у байтах потрібної пам'яті. Функція повертає вказівник типу `void` на виділений шматок пам'яті або `NULL`, якщо доступної пам'яті недостатньо. Щоб повернути вказівник на тип, відмінний від `void`, треба застосувати оператор явного перетворення типу (`cast-type*`). Наприклад, `int * pointer; pointer = (int*) malloc(100 * sizeof(int));`.

Приклад. У програмі (рис. 6.1) викликається функція `malloc` (рядок 8). Функція повертає адресу на виділену пам'ять. Ця адреса присвоюється змінній `p` типу вказівник на `int`. Далі в циклі комірки заповнюються значеннями (рядки 13-14). В кінці програми виводяться на екран адреси та значення, які містяться за цими адресами (рис. 6.2).

```

1  #include<stdio.h>
2  #include <stdlib.h>
3  int main()
4  {
5      int* p;
6      int n=15;
7          // запит на пам'ять
8      p = (int*) malloc (n * sizeof(int));
9      if(p==NULL){
10         printf("Not allocated, pointer p. \n"); }
11     else{printf("Allocated, pointer p. \n");
12         // заповнення виділеної пам'яті
13     for(int i=0;i<n;i++){
14         p[i] = i;
15         // виведення адрес елементів
16     for(int i=0;i<n;i++){
17         printf(" Address: %p \n", &p[i]);}
18         // виведення значень елементів
19     for(int i=0;i<n;i++){
20         printf("%d\n", p[i]);
21     }
22 }
23 }
```

Рисунок 6.1 – Робота з пам'яттю

```

Allocated, pointer p.
Address: 0x55757902e2a0
Address: 0x55757902e2a4
Address: 0x55757902e2a8
Address: 0x55757902e2ac
Address: 0x55757902e2b0
Address: 0x55757902e2b4
Address: 0x55757902e2b8
Address: 0x55757902e2bc
Address: 0x55757902e2c0
Address: 0x55757902e2c4
Address: 0x55757902e2c8
Address: 0x55757902e2cc
Address: 0x55757902e2d0
Address: 0x55757902e2d4
Address: 0x55757902e2d8
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
...Program finished with exit code 0

```

Рисунок 6.2 – Адреси та вміст за цими адресами

Динамічна пам'ять не може сама звільнити зайняту пам'ять, і наявний простір залишається навіть у кінці програми. Тому нам потрібно звільнити зарезервовану пам'ять, щоб інші програми могли її повторно використовувати. Функція `free(pointer)` звільняє динамічний розподіл пам'яті, створений за допомогою функції `malloc()`. Синтаксис функції: `void free(void *ptr);`.

Завдання для всіх варіантів

1. Розробити програму, в якій проводиться задана у варіанті обробка заданого тексту, який треба розмістити в динамічно виділеній пам'яті заданого у варіанті розміру.

Індивідуальне завдання

1. Варіанти завдань наведено в табл. К.1.

КОНТРОЛЬНІ ПИТАННЯ

1. Що означає динамічний розподіл пам'яті?
2. Які функції в мові С забезпечують управління пам'яттю?
3. Які проблеми можуть виникнути при роботі з динамічною пам'яттю?
4. Який специфікатор формату дозволяє виводити адреси?

Лабораторна робота № 7. Функції та макроси

Мета роботи: ознайомлення з особливостями використання макросів та функцій в програмах на мові C.

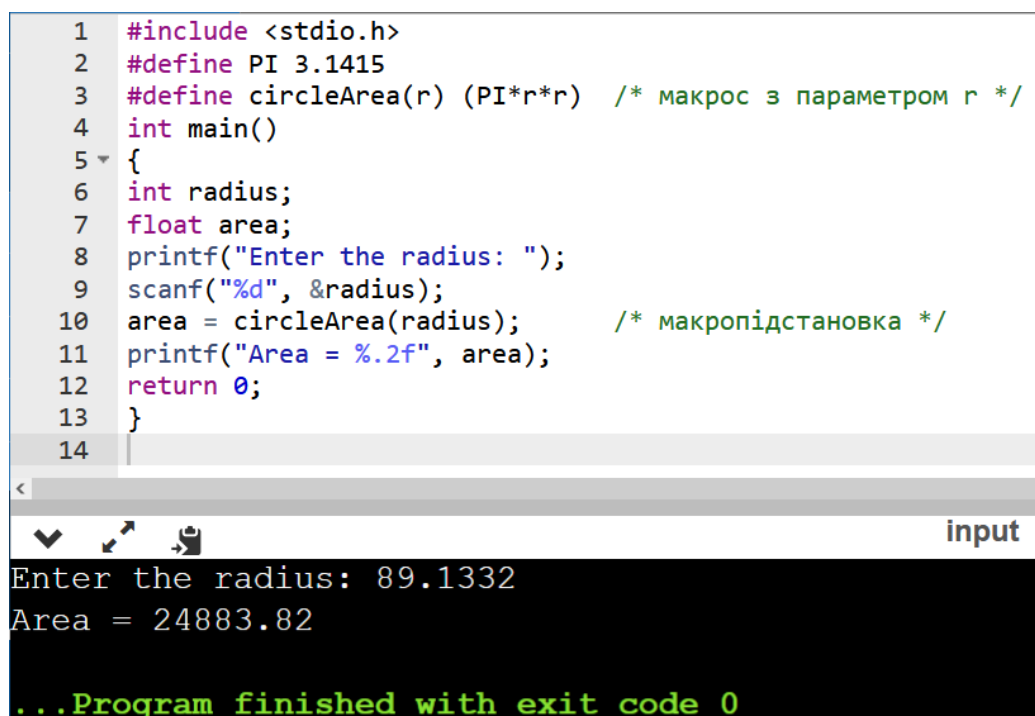
Коротка довідкова інформація

В сучасному програмуванні макроси використовуються не так активно, як раніше. Більшість програмістів схиляються до функцій. Проте в деяких ситуаціях використання макросів є доцільним. Загальні рекомендації, які враховують переваги та недоліки функцій та макросів, такі: макроси корисні, коли невеликий фрагмент коду використовується кілька разів у програмі; функції корисні, коли великий фрагмент коду повторюється кілька разів.

Макрос по суті – це фрагмент коду, якому дано ім'я. В мові C реалізується через директиву препроцесора `#define`.

Можна також визначити макроси з параметрами, які працюють як аналог виклику функції, так звані функціональні макроси.

В програмі (рис. 7.1) використано макрос та функціональний макрос для обчислення площі кола. В рядках 2 та 3 коду записано ці макроси, їх визначення. Рядок 10 коду препроцесор замінить на `area = 3.1415*radius*radius`.



```

1  #include <stdio.h>
2  #define PI 3.1415
3  #define circleArea(r) (PI*r*r) /* макрос з параметром r */
4  int main()
5  {
6  int radius;
7  float area;
8  printf("Enter the radius: ");
9  scanf("%d", &radius);
10 area = circleArea(radius); /* макропідстановка */
11 printf("Area = %.2f", area);
12 return 0;
13 }
14

```

input

Enter the radius: 89.1332

Area = 24883.82

...Program finished with exit code 0

Рисунок 7.1 - Приклад використання макросу

В програмі (рис. 7.2) використано макрос та функцію для знаходження максимального з двох цілих чисел.

```

1  #include<stdio.h>
2  #define max(a,b) (a>b)?a:b
3  int Max(int a, int b)
4  {
5      if (a>=b) return a;
6      else return b;
7  }
8  int main()
9  {
10     int x, y;
11     x=max(99,19);
12     printf(" x = %d", x);
13
14     y=Max(99,19);
15     printf(" y = %d ", y);
16 }
17

```

x = 99 y = 99

...Program finished with exit code 0

Рисунок 7.2 – Приклад використання макросу та функції

Завдання для всіх варіантів

1. Написати програму, яка реалізує індивідуальне завдання двома способами: за допомогою авторської функції; за допомогою макросу.

Індивідуальне завдання

2. Варіанти завдань наведено в табл. Л.1.

КОНТРОЛЬНІ ПИТАННЯ

1. Який спосіб передачі аргументів при виклику функції обрано в мові С?
2. Чим характеризується тип функції?
3. Що таке макроси?
4. Дайте порівняльний аналіз функцій та макросів, їх переваги та недоліки.
5. Яка директива препроцесора забезпечує використання макросу?

Лабораторна робота № 8. Бібліотечні функції

Мета роботи: ознайомлення з обробкою переривань за допомогою стандартної функції обробки сигналів `signal()`.

Коротка довідкова інформація

Деякі робочі середовища використовують сигнали для інформування запущених процесів про певні події. Ці події можуть бути пов'язані з помилками, які виконує програма, наприклад, неправильна арифметична операція. Також може виникнути якась виняткова ситуація, наприклад, запит на переривання програми. Сигнал може повідомляти про певну виняткову поведінку як в програмі, так і про асинхронну подію поза програмою.

Заголовок `<signal.h>` визначає дві функції `signal()` та `raise()`, також типи та макроси для обробки сигналів.

Функція `signal()` має синтаксис `void (*signal(int sig, void (*handler) (int))) (int);`.

При виклику функції треба задати значення `sig` в якості першого аргумента, вказівник на функцію - обробник `handler` в якості другого параметра.

Функція `signal()` використовується для визначення способу обробки сигналу, визначеного параметром `sig`. Параметр `handler` визначає спосіб, у який програма може обробляти сигнал. Це має бути одне з наступного:

- `SIG_DFL` (обробка за замовчуванням): сигнал обробляється дією за замовчуванням для цього конкретного сигналу;
- `SIG_IGN` (Ігнорувати сигнал): сигнал ігнорується, і виконання коду продовжиться, навіть якщо воно не має сенсу;
- функція - обробник: певна функція, визначена програмістом для обробки сигналу.

`SIG_DFL` або `SIG_IGN` встановлюється як поведінка обробки сигналу за замовчуванням під час запуску програми для кожного з підтримуваних сигналів.

В даній роботі пропонується розробити програму – обробник зовнішнього переривання, а саме, обробку натискання користувачем клавіш `CTRL-C`. Для цього треба застосувати один з макросів – `SIGINT`, який означає, що сигнал - інтерактивний, тобто генерується користувачем.

На рис. 8.1 показана програма, яка в циклі виводить слово `Nonstop`, причому цикл – нескінченний. Для його припинення користувач натискає клавіші `CTRL-C`, отримує повідомлення від програми, обирає варіант подальших дій (або закінчення циклу, або продовження). Результати наведено на рис. 8.2.

```

main.c
1  #include <stdio.h>
2  #include <unistd.h>
3  #include <stdlib.h>
4  #include <signal.h>
5  void handler(int sig)
6  {
7      char c;
8      signal(sig, SIG_IGN);
9      printf("External signal received\n");
10     printf("Do you really want to quit [y/n]?");
11     c = getchar();
12     if (c == 'y' || c == 'Y')
13         exit(0);
14     else
15         signal(SIGINT, handler);
16 }
17 void main(void)
18 {
19     signal(SIGINT, handler);
20     while (1)
21     {printf("Nonstop\n");
22       sleep(1);
23     }
24 }
25

```

Рисунок 8.1 – Використання функції signal()

```

Nonstop
Nonstop
Nonstop
Nonstop
^CExternal signal received
Do you really want to quit [y/n]?y
...Program finished with exit code 0

```

Рисунок 8.2 - Результати обробки зовнішнього переривання

Завдання для всіх варіантів

1. Написати програму для обробки зовнішнього переривання циклу.

Індивідуальне завдання

1. Варіанти завдань наведено в табл. М.1.

КОНТРОЛЬНІ ПИТАННЯ

1. Як підключити бібліотечні функції в мові C?
2. Наведіть приклади використання обробки сигналів в середовищі програмування.
3. Як відрізнити бібліотечні функції від інших в мові C?
4. Дайте приклади корисного застосування функцій роботи з часом та датами.
5. Що має містити опис будь-якої функції в документації по використанню?

Додаток А. Елементи мови С згідно стандарту

Таблиця А.1 – Елементи мови С

Лексичні елементи	Вирази. Константні вирази	Твердження та блоки	Декларації. Зовнішні визначення	Директиви препроцесора
Ключові слова Ідентифікатори Універсальні імена символів Константи Цілі константи Константи з плаваючою точкою Константи переліку Символьні константи Літерали - рядки Пунктуатори (роздільники) Імена заголовків Препроцесингові числа Коментарі	Первинні вирази Ідентифікатор Константа Рядок-літерал (Вираз) Загальний відбір Постфіксні оператори Індексація масиву [] Виклики функції () Елементи структури і об'єднання Оператори постфіксних інкремента і декремента ++, -- Складені літерали Унарні оператори Оператори префіксних інкремента і декремента ++, -- Оператори адреси та вмісту за адресою &, * Унарні арифметичні оператори Оператори sizeof та Alignof Оператори кастингу Мультиплікативні оператори Аддитивні оператори Оператори побітового зсуву Оператори відношення Оператори рівності Оператор побітового І Оператор побітового виключного АБО Оператор побітового включного АБО Логічний оператор І Логічний оператор АБО Умовний оператор Оператори присвоєння Оператор кома Константні вирази	Твердження з міткою Складене твердження (блок) Твердження -вирази та пусті твердження Твердження вибору Твердження if Твердження switch Твердження ітерації while do for Твердження швидкого переходу goto continue break return	Специфікатори класу зберігання typedef, extern, static, auto, register _Thread_local Специфікатори типу void, char, short, int, long, float, double, unsigned, signed, _Bool, _Complex Специфікатори структури та об'єднання struct, union Специфікатори переліку Enum Теги Специфікатори атомного типу _Atomic Визначення типу typedef Кваліфікатори типу const, restrict, volatile, _Atomic Специфікатори функції Inline, _Noreturn Специфікатори вирівнювання _Alignas Декларатори Прямий декларатор Декларатор масиву Декларатор функції Декларатор вказівника Імена (назви) типів Ініціалізація Статичні судження _Static_assert Зовнішні визначення Визначення функції Зовнішні визначення об'єкта	Умовна підстановка #if, #else, #elif, #ifdef, #ifndef, #endif, #if defined Включення початкового файлу #include Макроси Управління рядком Директива помилки Директива Pragma Пуста директива Зарезервовані імена макросів Оператор Pragma

Додаток Б. Індивідуальні завдання до ЛР № 1

Таблиця Б.1 – Завдання до ЛР № 1

Варіант № 1 Додати верхівку (у довільній формі) на ялинку.	Варіант № 2 Зробити пусті рядки між ярусами ялинки.	Варіант № 3 Зробити ялинку без стовбура, розташовану на квадратній підставці.	Варіант № 4 Додати одну прикрасу (символ Q) на праву половину ялинки.
Варіант № 5 Додати одну прикрасу (символ G) на ліву половину ялинки.	Варіант № 6 Намалювати кучугури біля ялинки.	Варіант № 7 Додати сніг у формі крапок.	Варіант № 8 Намалювати ялинку, яка стоїть в горщику (стовбура не видно).
Варіант № 9 Намалювати перевернуту ялинку.	Варіант № 10 Зсунути верхній ярус на одну позицію вправо.	Варіант № 11 Зсунути верхній ярус на одну позицію вліво.	Варіант № 12 Зробити видимою тільки праву половину ялинки.
Варіант № 13 Зробити видимою тільки ліву половину ялинки.	Варіант № 14 Зробити так, щоб стовбур ялинки малювався символом, який має обрати користувач, і який би відрізнявся від символу основного малюнка.	Варіант № 15 Намалювати ялинку в рамці.	Варіант № 16 Зробити ялинку вищою за рахунок зміни розміру нижньої частини стовбура.
Варіант № 17 Змістити ялинку на 2 позиції вправо.	Варіант № 18 Зробити обмеження на значення ширини та висоти ялинки, які може вводити користувач: ширина не більше 15-ти, а висота не більше 5-ти.	Варіант № 19 Зробити обмеження на набір символів для зображення ялинки, які може вводити користувач: не можна задавати літери f, z, k, g, цифри 2, 4, 5.	Варіант № 20 Зробити обмеження на набір символів для зображення ялинки, які може вводити користувач: можна задавати тільки літери x, m, s, v, цифри 1, 8, 0.
Варіант № 21 Змістити ялинку на 2 позиції вліво.	Варіант № 22 Зробити так, щоб при виведенні стовбур ялинки розташовувався симетрично відносно вертикальної вісі.	Варіант № 23 Додати один найнижчий ярус з більшою шириною та більшою висотою.	Варіант № 24 Додатково вивести з нового рядка загальну кількість використаних для малюнка символів.
Варіант № 25	Варіант № 26	Варіант № 27	Варіант № 28

Зробити так, щоб стовбур ялинки малювався символом, який додатково має обрати користувач, і який би відрізнявся від символа основного малюнка.	Додати одну прикрасу (символ М) на верхній ярус ялинки.	Додати одну прикрасу (символ W) на нижній ярус ялинки.	Зробити так, щоб при виведенні стовбур ялинки розташовувався симетрично відносно вертикальної вісі і щоб не було пробілів між символами в його зображенні.
Варіант № 29 Намалювати ялинку в рамці.	Варіант № 30 Зробити ялинку без стовбура, розташовану на прямокутній підставці.		

Додаток В. Таблиці формул фізики

Таблиця В.1 – Формули руху

№ п/п	формула	Значення формули	Величина	буква
1	$v = \frac{S}{t}$	Швидкість рівномірного. руху	швидкість	v
2	$v = v_0 + at$	Швидкість рівноприск. руху	Початкова шв.	v_0
3	$v = \frac{2\pi R}{T} = 2\pi R \nu = \omega R$	Швидкість колового руху	прискорення	a
4	$S = vt$	Шлях рівномірного руху	Доцентрове прискорення	a_0
5	$S = v_0 t + \frac{at^2}{2}$	Шлях рівноприск. руху	координата	x
6	$S = \frac{v + v_0}{2} t$	Шлях рівноприск. руху	Початкова координата.	x_0
7	$S = \frac{v^2 - v_0^2}{2a}$	Шлях рівноприск. руху	час	t
8	$a = \frac{v - v_0}{t}$	Прискорення	Кількість обертів	N
9	$x = x_0 + vt$	Рівняння руху рівномірного.	Кутова швидкість.	ω
10	$x = x_0 + v_0 t + \frac{at^2}{2}$	Рівняння рівноприск. руху	Дальність польоту	l
11	$a_0 = \frac{v^2}{R} = \frac{4\pi^2 R}{T^2} = 4\pi^2 R \nu^2 = \omega^2 R$	Доцентрове прискорення	Висота польоту	h
12	$T = \frac{1}{\nu} = \frac{t}{N}$	Період , частота	Період	T
13	$\omega = 2\pi \nu = \frac{2\pi}{T}$	Циклічна частота (кутова швидкість)	частота	ν
14	$l = v_0 \sqrt{\frac{2h}{g}}$	Дальність кинутого горизонтально	Прискорення вільного падіння	g
15	$l = \frac{v_0^2}{g} \sin 2\alpha$	Дальність кинутого під кутом	шлях	S
16	$h = \frac{v_0^2}{2g} \sin^2 \alpha$	Висота підняття під кутом	переміщення	s
17	$t = 2 \frac{v_0}{g} \sin \alpha$	Час польоту під кутом	Радіус кола	R

Таблиця В.2 – Формули механіки

№ п/п	формула	Значення формули	величина	буква
1	$g = \frac{Gm}{(R+h)^2}$	Прискорення вільного падіння	Маса Радіус планети	M R
2	$\rho = \frac{m}{V}$	Густина	Прискорення вільного падіння	g
3	$F_A = \rho g V$	Сила Архімеда	Сила об'єм	F V
4	$\frac{m_1}{m_2} = \frac{a_2}{a_1}$	Співвідношення між масами і прискореннями	Сила тертя	F _{тр}
5	$F_{тяж} = mg$	Сила тяжіння	Сила пружності	F _{gh}
6	$F = \frac{Gm_1m_2}{R^2}$	Сила всесвітнього тяжіння	Вага	P
7	$F_{mp} = \mu N$	Сила тертя	Жорсткість	k
8	$F_{mp} = kx$	Сила пружності	Видовження	x
9	$F = ma$	2 закон Ньютона	Коефіцієнт тертя	μ
10	$P = mg$ $P = m(g \pm a)$	Вага Перевантаження, недовантаження	Прискорення	a
11	$g = \sqrt{\frac{GM}{R+h}}$	Швидкість ШСЗ	Густина	ρ
12	$g = \sqrt{\frac{GM}{R}}$	1 космічна швидкість ШСЗ(h=0).	Плече сили	l
13	$p = m g$	Імпульс тіла	Імпульс тіла	p
14	$F = pS$	Сила тиску	Площа тиск	S p
15	$\eta = \frac{A_k}{A_i} 100\%$	К,К,Д,	К.к.д.	η
16	$E = \frac{m g^2}{2}$	Кінетична енергія	Енергія	E
17	$E = mgh$	Потенціальна енергія	Висота	h
18	$E = \frac{kx^2}{2}$	Потенціальна енергія Деформованого тіла	Потужність	P
19	$A = FS \cos \alpha$	Робота механічна	Механічна робота	A
20	$P = \frac{A}{t} = F g$	Потужність	Момент сили	M
21	$M = Fl$	Момент сили	гравітаційна стала	G

Таблиця В.3 – Формули термодинаміки

№ п/п	формула	Значення формули	величина	буква
1	$\mu = \mu, 10^{-3} \text{ кг / моль}$	Молярна маса	Молярна маса	μ
2	$P = \frac{1}{3} m_0 \cdot n \cdot v^2$	Основне р-ня МКТ	Маса молекули	m_0
3	$N = \frac{m}{\mu} N_A$	Кількість атомів	Кількість атомів	N
4	$\nu = \frac{N}{N_A} = \frac{m}{\mu}$	Кількість речовини в молях	Число Авогадро Концентрація	N_A n
5	$V = \sqrt{\frac{3kT}{m_0}}$	Середня квадратична швидкість молекул	Стала Больцмана	k
6	$PV = \frac{m}{\mu} RT$	Р-ня стану газу (Менделєєва-Клапейрона)	Абсолютна температура	T
7	$\frac{P_1 V_1}{T_1} = \frac{P_2 V_2}{T_2}$	Р-ня Клапейрона	тиск	p
8	$m_0 = \frac{m}{N} = \frac{\mu}{N_A}$	Маса одної молекули	об'єм	V
9	$U = \frac{3}{2} \frac{m}{\mu} RT$	Внутрішня енергія	Газова стала Внутрішня енергія	R U
10	$\bar{E} = \frac{3}{2} kT$	Енергія одної молекули	Енергія молекули	E
11	$Q = \Delta U + A$	1 закон термодинаміки	Робота газу	A
12	$\eta = \frac{Q_1 - Q_2}{Q_1} 100\%$	К.К.Д теплової машини	Зміна внутрішньої енергії	ΔU
13	$Q = rm \quad Q = qm$ $Q = \lambda m$ $Q = cm(t_2 - t_1)$	Теплоти пароутворення, згорання, плавлення, нагрівання	Кількість теплоти	Q
14	$n = \frac{N}{V}$	Концентрація	Відносне видовження	ε
15	$\varepsilon = \frac{\Delta l}{l_0} = \frac{l - l_0}{l_0}$	Відносне видовження	Абсолютне видовження	Δl
16	$\sigma = \frac{F}{S}$	Механічна напруга	Механічна напруга	σ
17	$\sigma = E\varepsilon$	Закон Гука	Модуль Юнга	E
18	$n = \frac{\sigma_M}{\sigma}$	Запас міцності	Запас міцності Границя міцності	n σ_M
19	$\sigma = \frac{F}{l}$	Коефіцієнт поверхневого натягу	Коефіцієнт поверхневого натягу	σ
20	$h = \frac{2\sigma}{\rho g r}$	Висота капіляра	Висота капіляра густина	h
21	$A = P\Delta V = P(V_2 - V_1)$	Робота газу	Радіус капіляра	R

Таблиця В.4 – Формули електрики

№ п/п	формула	Значення формули	величина	буква
1	$\sigma = \frac{q}{l}$	Лінійна густина заряду	Електричний заряд	q
2	$\sigma = \frac{q}{S}$	Поверхнева густина заряду	Довжина	l
3	$C = \frac{\epsilon \epsilon_0 S}{d}$	Ємність конденсатора	Площа	S
4	$E = \frac{F}{q} = \frac{kq}{r^2}$	Напруженість ел.поля	Відстань між пластинами	d
5	$\varphi = \frac{W}{q} = \frac{kq}{r}$	Потенціал	Напруженість	E
6	$E = \frac{U}{d}$	Зв'язок напруженості з напругою	Потенціал	φ
7	$C = \frac{q}{U} = \frac{q}{\varphi}$	Ємність	Відстань	r
8	$U = \varphi_1 - \varphi_2$	Напруга	Коефіцієнт пропорційності	k
9	$U = \frac{A}{q}$	напруга	Робота	A
10	$k = \frac{1}{4\pi\epsilon_0} \quad F = \frac{kq_1q_2}{r^2}$	Коефіцієнт пропорційності Закон Кулона	Напруга	U
11	$\epsilon = \frac{A_{сг}}{a}$	Електрорушійна сила лжерела	Електрична стала	ϵ_0
12	$I = \frac{\epsilon}{R+r}$	Закон Ома повного кола	Діелектрична проникність серед.	ϵ
13	$I = \frac{U}{R}$	Закон Ома ділянки	Опір	R
14	$P = IU = I^2 R$	Потужність ел. струму	Потужність	P
15	$I_{кз} = \frac{\epsilon}{r}$	Струм короткого замикання	Струм короткого замикання	$I_{кз}$
16	$A = IUt$	Робота струму	Внутрішній опір дж.	r
17	$I = \frac{q}{t}$	Сила струму	Час	t
18	$I = q_0 n S v$	Сила струму	Маса речовини в електролізі	m
19	$k = \frac{\mu}{e N_A n}$	Електрохімічний еквівалент	Електрохімічний еквівалент	k
20	$m = Ikt$	Закон електролізу	Валентність	n
21	$R = \frac{\rho l}{S}$ $R = R_0 (1 + \alpha \Delta t)$	Опір Залежність опору від температури	Питомий опір Температурний коефіцієнт опору	ρ α

Таблиця В.5 – Формули магнетизма

№ п/п	Формула	Значення формули	величина	буква
1	$B = \frac{M_{\max}}{IS}$	Магнітна індукція	Магнітна індукція	B
2	$B = \frac{F_{\max}}{Il}$	Магнітна індукція	Сила струму	I
3	$\Phi = BS \cos \alpha$	Магнітний потік	Магнітний потік	Φ
4	$B = \mu\mu_0 \frac{I}{2\pi d}$	Магнітна індукція нескінченного прямого провідника з струмом	Магнітна стала	μ_0
5	$B = \mu\mu_0 \frac{I}{2r}$	Магнітна індукція в центрі кругового витка	Магнітна проникність середовища	μ
6	$\Phi = LI$	Магнітний потік	Кількість витків	N
7	$M = BIS \sin \alpha$	Обертальний момент на рамку	Довжина провідника	l
8	$F_A = IBl \sin \alpha$	Сила Ампера	Електричний заряд	q
9	$F_L = Bqv \sin \alpha$	Сила Лоренца	Обертальний момент	M
10	$F = \frac{2\mu_0 \mu I_1 I_2 l}{4\pi r}$	Сила взаємодії двох провідників з струмом	Е.рс	E
11	$\varepsilon_i = -\frac{\Delta\Phi}{\Delta t}$	Закон електромагнітної індукції Для одного витка	Індуктивність	L
12	$\varepsilon_i = -N \frac{\Delta\Phi}{\Delta t}$	Закон електромагнітної індукції для соленоїда	Енергія магнітного поля	W
13	$\varepsilon_{ci} = -\frac{L\Delta I}{\Delta t}$	Закон електромагнітної самоіндукції	Сила Ампера	F _A
14	$L = \frac{\mu\mu_0 N^2 S}{l}$	Індуктивність котушки	Сила Лоренца	F _Л
15	$W = \frac{LI^2}{2}$	Енергія магнітного поля	Площа	S

Таблиця В.6 – Формули коливань

№ п/п	Формула	Значення формули	величина	буква
1	$T = \frac{t}{N}$	Період коливань	Період	T
2	$\nu = \frac{N}{t}$	Частота коливань	Частота	ν
3	$T = 2\pi\sqrt{\frac{l}{g}}$	Період математичного маятника	Фаза коливань	φ
4	$T = 2\pi\sqrt{\frac{m}{k}}$	Період пружинного маятника	Час	t
5	$\omega = 2\pi\nu = \frac{2\pi}{T}$	Циклічна частота	Кількість коливань	N
6	$\varphi = \omega t$	Фаза коливань	Заряд	q
7	$W = \frac{kx^2}{2}$	Енергія пружного деформованого тіла	Сила струму	I
8	$W = \frac{m\vartheta^2}{2}$	Енергія кінетична	Напруга	U
9	$x = x_m \cos(\omega t + \varphi_0)$	Рівняння коливального руху	Циклічна частота	ω
10	$\lambda = \vartheta t$	Довжина хвилі	Індуктивність	L
11	$T = 2\pi\sqrt{LC}$	формула Томсона	Ємність	C
12	$I_\delta = \frac{I_m}{\sqrt{2}} \quad U_\delta = \frac{U_m}{\sqrt{2}}$	Діючі значення струму і напруги	Амплітуда струму	I_m
13	$X_L = \omega L$	Індуктивний опір	Амплітуда напруги	U_m
14	$X_C = \frac{1}{\omega C}$	Ємнісний опір	Індуктивний опір	X_L
15	$R = \sqrt{R_a^2 + (x_L - x_C)^2}$	Повний опір в колі змінного струму	Ємнісний опір	X_C
16	$P = \frac{1}{2} I_m U_m = IU$	Потужність	Потужність	P
17	$\varepsilon_m = BS\omega$	Амплітуда ЕРС	Амплітуда ЕРС	E_m
18	$I = I_m \cos \omega t$ $U = U_m \cos \omega t$ $q = q \cos \omega t$	Миттєві значення струму, напруги заряду	Миттєві значення струму, напруги заряду	I U q
19	$W_m = \frac{LI^2}{2}$	Енергія магнітного поля котушки	Енергія магнітного поля котушки	W_L
20	$W = \frac{CU^2}{2}$	Енергія конденсатора	Енергія конденсатора	W_C

№ п/п	формула	Значення формули	величина	буква
1	$n = \frac{\sin \alpha}{\sin \beta}$	Закон заломлення	Збільшення лінзи	Γ
2	$\frac{1}{F} = \frac{1}{d} + \frac{1}{f}$	Формула лінзи	Оптична сила лінзи	D
3	$\Gamma = \frac{H}{h} = \frac{f}{d}$	Збільшення лінзи	Довжина хвилі	λ
4	$D = \frac{1}{F}$	Оптична сила лінзи	Стала дифракц. решітки	D
5	$d \sin \alpha = k \lambda$	Закон дифракційної решітки	Стала Планка	h
6	$\lambda = \frac{c}{\nu} = \frac{c}{f}$	Довжина хвилі	Період	T
7	$m = \frac{m_0}{\sqrt{1 - \frac{v^2}{c^2}}}$	Відносність маси	Частота	ν
8	$\frac{g}{1 + \frac{g_1 g_2}{c^2}} = \frac{g_1 + g_2}{1 + \frac{g_1 g_2}{c^2}}$	Закон додавання швидкостей	Робота виходу	A
9	$l = l_0 \sqrt{1 - \frac{v^2}{c^2}}$	Відносність довжини	Імпульс	p
10	$t = \frac{t_0}{\sqrt{1 - \frac{v^2}{c^2}}}$	Відносність часу	Маса	m
11	$E = mc^2$	Зв'язок енергії з масою	Енергія	E
12	$h\nu = A + \frac{mv^2}{2}$	Рівняння Ейнштейна	Швидкість світла	c
13	$p = mc = \frac{E}{c} = \frac{h\nu}{c} = \frac{h}{\lambda}$	Імпульс фотона	Довжина	λ
14	$m = \frac{E}{c^2} = \frac{h\nu}{c^2} = \frac{h}{c\lambda}$	Маса фотона	Час	t
15	$E = h\nu = \frac{hc}{\lambda}$	Енергія фотона	Період піврозпаду	T
16	$h\nu_{\min} = A$	Червона межа фотоефекту	Дефект маси	Δm
17	$N = N_0 2^{-\frac{t}{T}}$	Закон радіоактивного розпаду	Кількість протонів	Z
18	$\Delta m = Zm_p + Nm_n - (m_a - Zm_e)$	Дефект маси	Кількість нейтронів	N
19	$E_e = \Delta mc^2 = 931.5 \frac{\text{MeV}}{\text{aom}} \Delta m$	Енергія зв'язку ядра	Маса атома	m_a

Додаток Г. Індивідуальні завдання до ЛР № 2

Таблиця Г.1 – Завдання до ЛР № 2

[illegible]









Варіант № 25 Таблиця В.1 – формула 2 Таблиця В.2 – формула 18 Таблиця В.4 – формула 8 Таблиця В.6 – формула 4	Варіант № 26 Таблиця В.2 – формула 19 Таблиця В.4 – формула 9 Таблиця В.6 – формула 8 Таблиця В.7 – формула 1	Варіант № 27 Таблиця В.2 – формула 20 Таблиця В.3 – формула 11 Таблиця В.4 – формула 10 Таблиця В.6 – формула 9
Варіант № 28 Таблиця В.2 – формула 21 Таблиця В.4 – формула 11 Таблиця В.6 – формула 11 Таблиця В.7 – формула 4	Варіант № 29 Таблиця В.3 – формула 2 Таблиця В.4 – формула 12 Таблиця В.6 – формула 12 Таблиця В.7 – формула 16	Варіант № 30 Таблиця В.2 – формула 5 Таблиця В.3 – формула 3 Таблиця В.4 – формула 13 Таблиця В.6 – формула 15

Додаток Д. Індивідуальні завдання до ЛР № 3

Таблиця Д.1 – Завдання до ЛР № 3

Варіант № 1

Дані для занесення у файл:

	Тарас Шевченко <input type="text"/>
	Іван Франко <input type="text"/>
	Іван Котляревський <input type="text"/>
	Григорій Сковорода <input type="text"/>
	Володимир Винниченко <input type="text"/>
	Микола Гоголь <input type="text"/>
	Марія Матіос <input type="text"/>
	Валер'ян Підмогильний <input type="text"/>
	Василь Стус <input type="text"/>
	Леся Українка <input type="text"/>

Знайти і вивести на екран прізвища письменників, які починаються з заданої користувачем букви.

Варіант № 2

Дані для занесення у файл:

		
Cherry Blossom	Lily	Daisy
		
Poppy	Rose	Bluebell
		
Lotus	Forget Me Not	Aster
		
Violet	Peony	Bird of Paradise
		
Iris	Ixora	Tiger Lily

Знайти і вивести на екран всі назви квітів, довжина яких (у символах) менше вказаної користувачем.

Варіант № 3

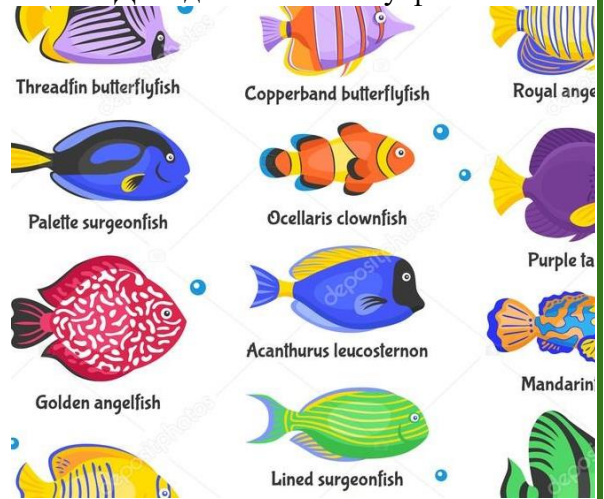
Дані для занесення у файл:

- 14.Cabbage
- 15.Chili
- 16.Peas
- 17.Capsicum
- 18.Lentil
- 19.Spinach
- 20.Onion
- 21.Brinjal
- 22.Asparagus
- 23.Carrot
- 24.Potato
- 25.Garlic

Знайти і вивести назви овочів, номери яких
(на рисунку) вкаже користувач.

Варіант № 4

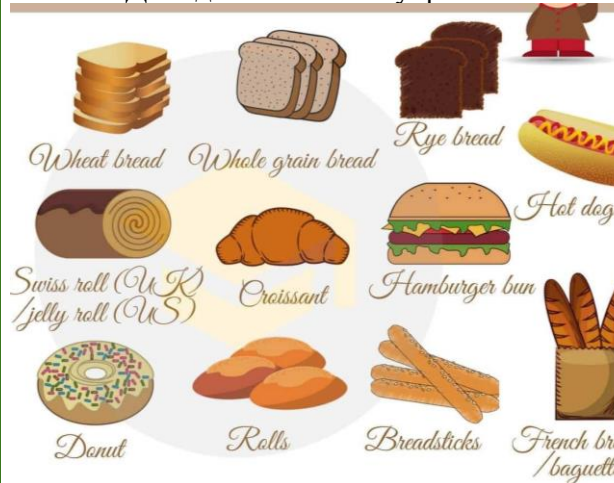
Дані для занесення у файл:



Знайти і вивести на екран назви рибок,
які містять вказану користувачем букву
щонайменше двічі.

Варіант № 5

Дані для занесення у файл:



Вивести на екран назви хлібних виробів,
які містять дві задані користувачем букви.

Варіант № 6

Дані для занесення у файл:



Вивести на екран назви дерев, які містять
задані користувачем дві голосні букви.

Варіант № 7

Дані для занесення у файл:

- Edam Deepheart
- Taiana Sunbane
- Roberd Swift dust
- Claye Springglow
- Malcalm Springbrace
- Arana Marshflare
- Jaleah Lightmoon
- Adriel Shadowgem
- Kaze Shadowbow

Вивести на екран імена, які закінчуються
на задану користувачем букву.

Варіант № 8

Дані для занесення у файл:



Знайти і вивести на екран породи котів,
які містять вказані користувачем дві
букви початку назви.

Варіант № 9

Дані для занесення у файл:

Sapphire	Gizmo	Onyx
Rasta	Bonnie	Hiro
Linux	Sierra	Siri
Flynn	Lexi	Nala
Remy	Skye	Spunky
Spaz	Kinko	Zues

Знайти і вивести на екран всі котячі імена,
довжина яких (у символах) дорівнює
вказаній користувачем.

Варіант № 10

Дані для занесення у файл:

Fruitella	Lemon Drop
Gummy	Lifesavers
Green Lollipop	Licourice
Gumdrops	Lindor
Ginger	Loly Pop
Grand	Licquorice
Gummy Worms	Lemon Gummies
Hershey	Marathon

Знайти і вивести на екран всі назви
цукерок, довжина яких (у символах)
більше вказаної користувачем.

Варіант № 11

Дані для занесення у файл:



Знайти і вивести на екран назву кави за номером (з картинки), який введе користувач.

Варіант № 12

Дані для занесення у файл:



Знайти і вивести на екран назву фільму, яка відповідатиме введеному користувачем номеру на афіші.

Варіант № 13

Дані для занесення у файл:



Знайти і вивести на екран породи собак, які починаються та закінчуються заданими користувачем буквами.

Варіант № 14

Дані для занесення у файл:

1. Cinderella (birthday girl)
2. Gretel
3. Snow White
4. Little Red Ridinghood
5. Goldilocks
6. Princess and the Pea
7. Tinkerbell
8. Sleeping Beauty
9. Little Bo Peep
10. Rapunzel
11. Beauty
12. Dancing Princess

Знайти і вивести на екран імена героїв казок, які відповідатимуть введеним користувачем номерам в списку.

Варіант № 15

Дані для занесення у файл:

LIST OF FAMOUS PAINTERS

- JOAN MIRÓ
- MARC CHAGALL
- WILLEM DE Kooning
- FRANCIS BACON
- GUSTAV KLIMT
- EDWARD HOPPER
- AMEDEO MODIGLIANI
- MARC CHAGALL
- RENÉ MAGRITTE
- ROY LICHTENSTEIN

Знайти і вивести на екран прізвища художників, які не містять заданої користувачем букви.

Варіант № 16

Дані для занесення у файл:



Martha Washington
Winston Churchill
Albert Einstein
Blaise Pascal
C S Lewis

Mark Twain
Hans C. Anderson
Clara Barton
Beatrix Potter
Abigail Adams

Знайти і вивести на екран імена діячів за заданим користувачем прізвищем.

Варіант № 17

Дані для занесення у файл:

 Water	 Coffee	 Tea	 Milk
 Milkshakes	 Soya Milk	 Apple Juice	 Soft drink
 Lemonade	 Iced tea	 Smoothies	 Soda wat
 Almond milk	 Orange juice	 Grape juice	 Herbal te

Знайти і вивести на екран назви напоїв, які складаються з заданої користувачем кількості букв.

Варіант № 18

Дані для занесення у файл:

1. cup
2. electric kettle
3. fork
4. jug
5. kettle
6. knife
7. pan
8. pepper
9. plate
10. salt
11. saucer
12. spoon
13. stewpan
14. sugar
15. teapot
16. tea cup

Знайти і вивести на екран всі назви кухонних предметів, довжина яких менша за задану користувачем.

Варіант № 19

Дані для занесення у файл:



Знайти і вивести на екран прізвища письменників, ім'я та прізвище яких починаються з заданих користувачем двох букв.

Варіант № 20

Дані для занесення у файл:



Знайти і вивести назву метелика, розташованого на рисунку, за вказаними користувачем координатами (рядок, стовпчик).

Варіант № 21

Дані для занесення у файл:



Вивести на екран назви жанрів кінофільмів у кількості, заданій користувачем.

Варіант № 22

Дані для занесення у файл:



Знайти і вивести на екран назви інструментів, які починаються з заданої користувачем букви.

Варіант № 23

Дані для занесення у файл:

AEROBICS
FOOTBALL
HORSEBACK RIDING
BASKETBALL
ROWING
SKATEBOARDING
MOTOR RACING
ICE SKATING
BOXING
ROCK CLIMBING
CRICKET
FENCING
POLO
TABLE TENNIS
RUGBY

Знайти і вивести на екран всі назви видів спорту, довжина яких (у символах) більше вказаної користувачем.

Варіант № 24

Дані для занесення у файл:



Знайти і вивести на екран всі назви вулканів, довжина яких (у символах) дорівнює вказаній користувачем.

Варіант № 25

Дані для занесення у файл:



Знайти і вивести на екран назви чаю, які містять вказану користувачем букву всередині назви.

Варіант № 26

Дані для занесення у файл:



Вивести на екран повідомлення про наявність в масиві назви марки авто, заданої користувачем.

Варіант № 27

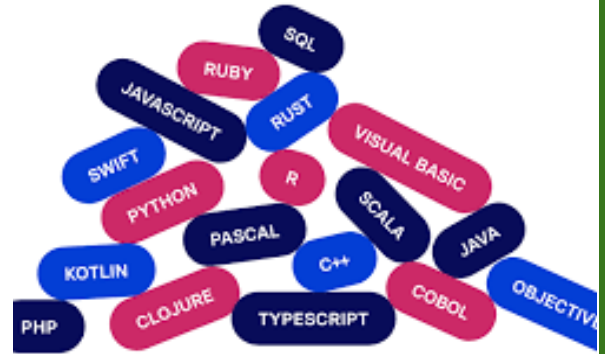
Дані для занесення у файл:



Вивести на екран назви птахів, номери яких в масиві задасть користувач.

Варіант № 28

Дані для занесення у файл:



Знайти і вивести на екран всі назви мов програмування, довжина яких (у символах) дорівнює вказаній користувачем.

Варіант № 29

Дані для занесення у файл:

Amaretto	Dulce de Leche	Peach
Banana	Dutch Honey	Peanut Butter
Blackberry	Eggnog	Pepsi
Black Raspberry	Fudge Chocolate	Pina Colada
Blueberry	Grape	Pineapple

Знайти і вивести назву смаку, розташованого на рисунку, за вказаними користувачем координатами (рядок, стовпчик).

Варіант № 30

Дані для занесення у файл:



Знайти і вивести назви сузір'їв у кількості, вказаній користувачем.

Додаток Е. Індивідуальні завдання до ЛР № 4

Таблиця Е.1 – Завдання до ЛР № 4

Варіант № 1 Варіант № 2 з ЛР №3.	Варіант № 2 Варіант № 1 з ЛР №3.
Варіант № 3 Варіант № 4 з ЛР №3.	Варіант № 4 Варіант № 3 з ЛР №3.
Варіант № 5 Варіант № 6 з ЛР №3.	Варіант № 6 Варіант № 5 з ЛР №3.
Варіант № 7 Варіант № 8 з ЛР №3.	Варіант № 8 Варіант № 7 з ЛР №3.
Варіант № 9 Варіант № 10 з ЛР №3.	Варіант № 10 Варіант № 9 з ЛР №3.
Варіант № 11 Варіант № 12 з ЛР №3.	Варіант № 12 Варіант № 11 з ЛР №3.
Варіант № 13 Варіант № 14 з ЛР №3.	Варіант № 14 Варіант № 13 з ЛР №3.
Варіант № 15 Варіант № 16 з ЛР №3.	Варіант № 16 Варіант № 15 з ЛР №3.
Варіант № 17 Варіант № 18 з ЛР №3.	Варіант № 18 Варіант № 17 з ЛР №3.
Варіант № 19 Варіант № 20 з ЛР №3.	Варіант № 20 Варіант № 19 з ЛР №3.
Варіант № 21 Варіант № 22 з ЛР №3.	Варіант № 22 Варіант № 21 з ЛР №3.
Варіант № 23 Варіант № 24 з ЛР №3.	Варіант № 24 Варіант № 23 з ЛР №3.
Варіант № 25 Варіант № 26 з ЛР №3.	Варіант № 26 Варіант № 25 з ЛР №3.
Варіант № 27 Варіант № 28 з ЛР №3.	Варіант № 28 Варіант № 27 з ЛР №3.
Варіант № 29 Варіант № 30 з ЛР №3.	Варіант № 30 Варіант № 29 з ЛР №3.

Додаток Ж. Індивідуальні завдання до ЛР № 5

Таблиця Ж.1 – Завдання до ЛР № 5

Варіант № 1

Дані для занесення у файл:

Pollutant	Emissions		Typical Concentration	
	Natural (10 ⁶ t/y)	Anthropogenic (10 ⁶ t/y)	Background (ppm)	Polluted Air (ppm)
SO ₂	40	150	0.0002	0.2
H ₂ S	100	3	0.0002	–
CO	33	1077	0.1	40-70
NO _x (as NO)	430	83	<0.002	–
NO _x (as NO ₂)	658	127	<0.004	0.2
NH ₃	1160	12	0.01	0.02
N ₂ O	18	12	0.31	–
Volatile Organics	200	186	<0.001	–
CH ₄	1600	321	1.7	2.5

Користувач задає назву забруднювача. Вивести дані про цей забруднювач з відповідного рядка таблиці.

Варіант № 2

Дані для занесення у файл:







Назва	Частина світу	Площа	Кількість населення	Дата утворення держави	Державний устрій
Аргентина	Південна Америка	2767	37032	01.01.1826	Парламентська республіка
Ангола	Африка	1246,7	13134	11.11.1975	Президентська республіка
Албанія	Європа	28,7	3582	28.11.1912	Президентська республіка
Австрійська Республіка	Європа	83,8	8080	12.11.1918	Парламентська республіка
Австралія	Австралія і Океанія	7636	19138	01.01.1901	Конституційна монархія
Алжирська Народна Демократична Республіка	Африка	2381,7	33190	05.07.1962	Президентська республіка
Антигуа і Барбуда	Північна і Центральна Америка	0,4	69	01.11.1981	Конституційна монархія
Азербайджан	Європа	86,6	7868	30.08.1991	Президентська республіка

Користувач задає державний устрій. Вивести відповідні країни.

Варіант № 3

Дані для занесення у файл:

SANDWICHES

 <p>HAMBURGER 250 Calories 9g Fat 31g Carbs 12g Protein</p>		<p>CHEESEBURGER 300 Calories 13g Fat 32g Carbs 15g Protein</p> 
 <p>DOUBLE CHEESEBURGER 450 Calories 24g Fat 34g Carbs 25g Protein</p>		<p>McDOUBLE 400 Calories 20g Fat 33g Carbs 22g Protein</p> 

Користувач задає кількість протеїну у страві. Вивести відповідні назви сендвічів.

Варіант № 4

Дані для занесення у файл:

Роки спостережень	Обсяги викидів забруднюючих речовин		
	усього, тис. т	у тому числі	
		стаціонарними джерелами	пересувними джерелами
2009	84,2	18,6	65,6
2010	87,5	18,4	69,1
2011	86,8	19,0	67,8
2012	85,7	18,5	67,2
2013	82,5	17,2	65,3

Користувач задає рік спостережень. Вивести відповідний рядок таблиці.

Варіант № 5

Дані для занесення у файл:

THE CLASSICS		PREMIUM	
Italiano	8" 6.49 15" 10.99	Cleveland Burger	8" 6.99 15" 11.99
Provolone & Mozzarella Cheese Blend, Ham, Salami, Pepperoni, Onions, Banana Peppers, Lettuce, Tomatoes & Italian Dressing		Provolone & Mozzarella Cheese Blend, Seasoned Steak Patties, Salami, Onions, Lettuce, Tomatoes & Creamy Italian Dressing	
Club	8" 6.49 15" 10.99	Caprese	8" 6.99 15" 11.99
Provolone & Mozzarella Cheese Blend, Ham, Bacon, Turkey, Lettuce, Tomatoes & Mayo		Fresh Tomato Slices, Fresh Basil & Fresh Mozzarella, baked atop our Garlic Bread & finished off with Balsamic Glaze	

Користувач задає назву піци. Вивести відповідну ціну та склад з таблиці.

Варіант № 6

Дані для занесення у файл:

	carat	cut	color	clarity	depth	table	price
1	0.52	Ideal	D	VS2	61.4	56	1664
2	0.5	Very Good	F	SI1	62.3	60	1250
3	0.61	Ideal	G	VVS2	61.6	54	2242
4	0.36	Premium	G	VS2	62.5	58	756
5	0.7	Very Good	E	VS2	63.5	54	2889
6	0.56	Ideal	F	VS1	61.7	56	2016

Користувач задає колір (color) цінного каменю. Вивести відповідні ціни та прозорості (clarity).

Варіант № 7

Дані для занесення у файл:
















	1-13 years*		14 years and above*	
Vitamins	RDA	UL	RDA	UL
Vit A	1000-2000 IU	2000-6000 IU	1000-3000IU	9000-10000IU
Vit B1 (Thiamine)	0.5-0.9 mg	NE	1-1.4 mg	NE
Vit B2 (Riboflavin)	0.5-0.9 mg	NE	1-1.4 mg	NE
Vit B3 (Nicotinamide)	6-12 mg	10-20 mg	14-18 mg	30-35 mg
Vit B5 (Pantothenate)	2-4 mg	NE	5-7 mg	NE
Vit B6 (Pyridoxine)	0.5-1 mg	30-60 mg	1.2-2.0 mg	80-100 mg

Користувач задає назву вітаміна та вікову групу. Вивести відповідну RDA.

Варіант № 8

Дані для занесення у файл:

Top 10 Worldwide Grossing Mobile, Handheld and PC Games By Platform Q1 2021

Rank	iOS App Store & Google Play	Nintendo Switch Lite	Steam (2020)
1	 ROBLOX Roblox, United States	 Super Mario 3D World + Bowser's Fury Nintendo, Japan	 Counter-Strike: Global Offensive Valve Corp., US
2	 Genshin Impact miHoYo, China	 Monster Hunter Rise Capcom, Japan	 Dota 2 Valve Corp., US
3	 Coin Master Moon Active, Israel	 Mario Kart 8 Deluxe Nintendo, Japan	 Grand Theft Auto V Rockstar Games, US
4	 Pokémon GO Niantic, United States	 Animal Crossing: New Horizons Nintendo, Japan	 PlayerUnknown's Battlegrounds PUBG Corp., South Korea
5	 Honour of Kings Tencent, China	 Super Smash Bros. Ultimate Nintendo, Japan	 Cyberpunk 2077 CD Projekt, Poland

Користувач задає ранг гри. Вивести повну інформацію для цього рангу.

Варіант № 9

Дані для занесення у файл:

Безалкогольні напої		
	PLN	TRY
МІН. ВОДА «BONAQUA»	500	20
МІН. ВОДА «BONAQUA»	1000	25
МІН. ВОДА «ПОЛЯНА КВАСОВА» (СКЛО)	500	60
ШВЕПС	1000	45
КОКА КОЛА (СКЛО)	250	30
КОКА КОЛА	500	35
КОКА КОЛА	1000	45
СІК «РІЧ» В АСОРТИМЕНТІ	1000	70
УЗВАР	1000	50
Премі		
	350 PLN	1000 PLN
АПЕЛЬСИНОВИЙ	105	350
ЯБЛУЧНИЙ	60	200
МОРКВ'ЯНИЙ	75	250

Користувач задає об'єм напою. Вивести його назву та ціну.

Варіант № 10

Дані для занесення у файл:

Film Title	Released	Studio	Worldwide Gross	Domestic Gross	International Gross	Budget
The Lion King (2019 remake)	2019	Disney	\$1,657,870,986	\$543,638,043	\$1,114,232,943	\$260,000,000
Frozen II	2019	Disney	\$1,450,026,933	\$477,373,578	\$972,653,355	\$150,000,000
Frozen	2013	Disney	\$1,281,019,275	\$400,953,009	\$880,066,266	\$150,000,000
Incredibles 2	2018	Disney Pixar	\$1,243,089,244	\$608,581,744	\$634,507,500	\$200,000,000
Minions	2015	Universal	\$1,159,444,662	\$336,045,770	\$823,398,892	\$74,000,000

Користувач задає рік виходу фільму. Вивести назви та студії.

Варіант № 11






Дані для занесення у файл:

Бренд	Місце в топ-500	Країна	Сфера діяльності	Вартість бренду, \$ млн	Зміна рік до року, %
Amazon 	1	США	E-commerce	187,9	+24,6
Walmart 	11	США	FMCG	67,9	+10,4
Home Depot 	22	США	DIY	47,1	+39,4
Starbucks 	31	США	Food	39,3	+21,1
Tmall 	35	Китай	E-commerce	36,4	н/д
McDonald's 	43	США	Food	31,5	+26,6

Користувач задає країну. Вивести назву бренда та його вартість.

Варіант № 12

Дані для занесення у файл:

Drink	milk	coffee	green tea	sake	champagne
Emoji					
Calories	149.0	2.0	2.5	39.0	89.0
Carbohydrates (g)	11.7	0.0	0.0	1.5	1.6
Total Sugar (g)	12.3	0.0	0.0	0.0	0.0
Protein (g)	7.7	0.3	0.5	0.1	0.4
Total Fat (g)	8.0	0.0	0.0	0.0	0.0
Total Fiber (g)	0.0	0.0	0.0	0.0	0.0
Cholesterol (mg)	24.4	0.0	0.0	0.0	0.0

Користувач задає назву напою. Вивести калорії та цукор.

Варіант № 13

Дані для занесення у файл:

United States			
Shipping Method	Costs*	Region	Shipping Time
Standard Shipping (Orders \$35 and over)	FREE	United States (All States)	7-12 business days <i>Rural & remote areas may take an additional 3 days</i>
Standard Shipping (Orders under \$35)	\$5	United States (All States)	7-12 business days <i>Rural & remote areas may take an additional 3 days</i>
International			
Shipping Method	Costs*	Region	Shipping Time
Standard Shipping (Orders \$75 and over)	FREE	International	15-30 business days <i>Rural & remote areas may take an additional 5 days</i>
Standard Shipping (Orders under \$75)	\$7	The United Kingdom	10-20 business days <i>Rural & remote areas may take an additional 5 days</i>

Користувач задає метод відправки. Вивести інформацію про час доставки.

Варіант № 14

Дані для занесення у файл:

Зведені продажі					
Дата (місяць)		Січень	Лютий	Березень	Загальний підсумок
Живлення	Продукт	Одиниці (сума)			
▼ Електричне	Велосипеди	476	354	392	1 222
	Мопеди	387	312	211	910
	Скейтборди	251	315	467	1 033
	Електричне у підсумку	1 114	981	1 070	3 165
▼ Ручне	Велосипеди	302	219	464	985
	Мопеди	309	419	293	1 021
	Ручне у підсумку	611	638	757	2 006
Загальний підсумок		1 725	1 619	1 827	5 171

Користувач задає продукт. Вивести кількість продажів у березні.

Варіант № 15




Дані для занесення у файл:

Items	Purchase Date	Units sold	Unit price	Revenue	Cost
Drinks	10/06/2014	4,382.00	47.45	\$207,925.90	\$139,303.78
Desk	22/12/2015	6,808.00	651.21	\$4,433,437.68	\$3,573,927.68
Grain	13/04/2015	5,528.00	205.70	\$1,137,109.60	\$647,384.08
Snacks	24/10/2013	9,369.00	152.58	\$1,429,522.02	\$912,915.36
Snacks	16/07/2012	8,149.00	152.58	\$1,243,374.42	\$794,038.56

Користувач задає тип продукції. Вивести дату покупки та дохід.

Варіант № 16

Дані для занесення у файл:

	 AWS	 AZURE	 GCP
LAUNCHING YEAR	2004	2010	2008
AVAILABILITY	84 availability zone & 24 Geographical locations	60+ region across all over the country	24 region & 74 total Zones
SERVICES	212+ SERVICES	200+ SERVICES	100+ Services
CLOUD SHARE	33% of the Market	21% of the Market	8% of the Market
COMPUTE ENGINE	EC2 (Elastic Compute System)	Virtual Machine	Compute Engine
NETWORKING	VIRTUAL PRIVATE CLOUD	VIRTUAL NETWORK (VNET)	CLOUD VIRTUAL NETWORK

Користувач задає назву хмарної платформи. Вивести інформацію про неї.

Варіант № 17

Дані для занесення у файл:

University Name	Entry Standards	Student Satisfaction	Research Quality	Graduate Prospects	Overall Score
Cambridge	600 	4.17 	3.33 	89.0 	1000 
Oxford	574 	4.17 	3.34 	86.7 	998 
London School of Economics	518 	3.89 	3.35 	78.5 	940 
Imperial College London	567 	4.15 	3.36 	90.8 	938 
St Andrews	520 	4.23 	3.13 	83.2 	927 

Користувач задає назву університету. Вивести інформацію про нього.

Варіант № 18

Дані для занесення у файл:

РЕПЕРТУАР ЛИПЕНЬ	
початок о 18:00	ВЕЛИКА СЦЕНА
04 липня	№ 13 <small>Р. Куні</small>
11 липня	ЗДРАСТУЙТЕ, Я ВАША ТІТОНЬКА <small>Б. Томас</small>
ЗАКРИТТЯ 75-ГО ТЕАТРАЛЬНОГО СЕЗОНУ	
початок о 12:00	ВИСТАВИ ДЛЯ ДІТЕЙ
04 липня	ПОВЕЛИТЕЛЬ ЧОРНОГО ЛІСУ <small>В. Шершун</small>
11 липня	МАУГЛІ <small>Р. Кіплінг</small>
початок о 18:30	МАЛА СЦЕНА
02 липня	ТОЙ, ЩО ВІДЧИНЯЄ ДВЕРІ <small>Н. Неждана</small>
09 липня	КРАЩЕ ЗАЛИШАЙСЯ МЕРТВИМ <small>К. Ветлінгер</small>

Користувач задає дату. Вивести назву вистави та сцену.

Варіант № 19

Дані для занесення у файл:

	Number of species	Number of individuals	% of individuals	Total weight (kg)
Mormyridae	19	1198	20.7	219
<i>Hyperopisus bebe</i>	1	166		49
<i>Mormyrus</i>	3	180		47
<i>Mormyrops</i>	3	122		55
<i>Campylomormyrus</i>	1	366		38
<i>Marcusenius</i>	4	292		27
<i>Hippopotamyrus</i>	3	56		3
others	4			
Alestidae	8	2103	36.3	136
<i>Hydrocynus</i>	2	28		11
<i>Alestes</i>	2	1447		100

Користувач задає кількість видів (number of species) африканських риб. Вивести відповідні назви.

Варіант № 20

Дані для занесення у файл:

Зведена таблиця успішності учнів за 2019/2020 навчальний рік

Школа I ступеня (3-4 класи)				
Рівень знань	Кількість учнів	%	Всього учнів	
Початковий рівень	0	0%	28	
Середній рівень	4	14,29%		
Достатній рівень	12	42,86%	Якість	85,71%
Високий рівень	12	42,86%	Успішність	100%

Школа II ступеня				
Початковий рівень	2	2,08%	96	
Середній рівень	11	11,46%		

Користувач задає рівень знань та ступінь школи. Вивести кількість учнів абсолютну та у відсотках.

Варіант № 21

Дані для занесення у файл:

<div>  Departures <div>Local time 09:46:31</div> </div>					
Terminal	Flight	Destination	Time	Gate	Remark
2	T01289	TOKYO	09:25	G18	DEPARTED
1	PA350	PRAGUE	09:40	F10	DEPARTED
1	PR1052	PARIS	09:50	C22	BOARDING
1	ML2115	MELBOURNE	10:05	C14	BOARDING
2	MA1165	MADRID	10:10	D9	BOARDING
1	BC643	BARCELONA	10:35	F20	BOARDING
1	NC2247	DUBAI	10:45	H17	CHECK-IN
1	OL1441	OSLO	10:50	B4	CHECK-IN
2	ZR345	ZURICH	11:20	E8	CHECK-IN
1	LD1216	LONDON	11:35	G10	SCHEDULED
1	CL1201	CLASCOM	12:10	B16	CANCELLED

Користувач задає номер терміналу. Вивести дані про відповідні рейси.

Варіант № 22

Дані для занесення у файл:

Таблиця 1. Кількість рослин після появи сходів, шт./м²

Варіант досліджу		Роки			
		2015	2016	2017	2018
Глибина внесення добрива NP (азот – фосфор)	0 см (поверхнево)	8,20	7,91	7,44	7,68
	5 см	8,14	7,86	7,35	7,68
	10 см	8,02	7,59	7,38	7,70
	15 см	7,83	7,40	7,23	7,59
Тип азотного добрива	аміачна селітра	8,09	7,62	7,36	7,65
	сечовина	8,00	7,75	7,34	7,68

Користувач задає рік. Вивести результати дослідів.

Варіант № 23

Дані для занесення у файл:

Сорт	Вегетаційний період (від посадки), діб	Забарвлення м'якуша	Особливості
Ранні			
Вардаман (Vardaman)	90–95	помаранчевий	середня солодкість, розсіпчастий, компактне розташування бульб
Орлеанс (Orleans)	90–100	оранжевий	урожайний
Ковінгтон (Covington)	90–100	помаранчевий	вирівняні бульби, компактне розташування бульб
Середні			
Адмірал	100–110	оранжевий	урожайний, високий вміст β-каротину, цукрів
Боніта (Boniato)	110	білий	урожайний, вирівняні бульби

Користувач задає один ранній сорт та один середній сорт. Вивести дані про ці сорти з таблиці.

Варіант № 24

Дані для занесення у файл:

СОРТИ ЗА ІНТЕНСИВНІСТЮ	НАЗВА СОРТУ	НОРМА ВИСІВУ, МЛН ШТ./ГА	ЗА УМОВИ ПОСУХИ
ВИСОКО-ІНТЕНСИВНІ (АДАПТИВНІ)	РІВЕРО	3,5-4,5	5,0
	РЖТ РЕФОРМ	4,0-5,0	5,5
	РЕБЕЛЛ	3,5-5,0	5,5
	МЕСКАЛЬ	4,0-5,0	5,5
ВИСОКО-ІНТЕНСИВНІ (ПОСУХОСТІЙКІ)	ТУРАНДОТ	3,0-4,5	5,0
	ЮЛІЯ	3,0-4,0	4,5
ВИСОКО-ІНТЕНСИВНІ (УНІВЕРСАЛЬНІ)	ФРАНЦ	4,0-4,5	5,0
	КОЛОНІА	4,5-5,5	6,0
УНІВЕРСАЛЬНІ (ПЛАСТИЧНІ)	ПРАКТИК	3,5-5,5	6,0
	МУЛАН	3,5-4,5	5,0

Користувач задає назву сорту. Вивести дані про цей сорт.

Варіант № 25

Дані для занесення у файл:

Таблиця 1. - ЕКОЛОГО-САНІТАРНИЙ СТАН ДНІПРОВСЬКОЇ ВОДИ

Інгредієнти	Міні-мальний	Макси-мальний	Середній	Категорія якості
Гідрофізичні показники				
Зважені частки, мг/дм ³	2,80	148,62	73,8	дуже чиста
Кольоровість, град	3,72	8,34	5,91	слабо забруднена
Прозорість, см	1,52	14,64	7,43	слабо забруднена
Гідрохімічні показники, мг/дм³				
No ₂	0,002	0,508	0,0035	слабо забруднена
No ₃	0	5,85	2,92	слабо забруднена

Користувач задає тип показника. Вивести дані про інгредієнти.

Варіант № 26

Дані для занесення у файл:

Культура	Вегетаційний період, днів	Необхідна сума активних температур, °С	Необхідна кількість вологи на 1м сухої речовини, г	Найбільш сприятливі ґрунти
Ярова пшениця	90-120	1200-1700	513	Чорноземи, каштанові
Озима пшениця	250	1150-1500	575	Чорноземи
Кукурудза	85-150	211-2900	368	Чорноземи, каштанові
Жито	140-180	1000-1200	610	Дерново-підзолисті
Льон-довгунець	75-90	950-1030	910	Дерново-підзолисті
Цукровий буряк	150-180	2200-2400	397	Чорноземи
Картопля	70-120	1200-1800	637	Дерново-підзолисті

Користувач задає назву культури. Вивести дані про цю культуру.

Варіант № 27

Дані для занесення у файл:

дані для занесення у файл.

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
100	A	5 ВІДМІННО – відмінне виконання лише з незначними помилками	ЗАРАХОВАНО
90			
89	B	4 ДУЖЕ ДОБРЕ – дозволяється перескладання на більш високу оцінку (A)	
82			
81	C	4 ДОБРЕ – перескладання на більш високу оцінку (A) не дозволяється.	
74			
73	D	3 ЗАДОВІЛЬНО – дозволяється перескладання на більш високу оцінку (але не вище C)	
64			
63	E	3 ДОСТАТНЬО – перескладання на більш високу оцінку не дозволяється	
60			

Користувач задає оцінку ECTS. Вивести інформацію про цю оцінку.

Варіант № 28

Дані для занесення у файл:

Країна	Тривалість повної середньої освіти (років) і її структура	Кількість навчальних днів		
		у віці 7 років	у віці 10 років	пізніше
Австрія	12–13 (4+4+4)	180 — 214	180 — 214	180
Англія та Уельс	13 (6+3+4)	190	190	190
Бельгія	12–13 (6+2+4)	182	182	182
Греція	12 (6+3+3)	175	175	175
Данія	12–13 (9+3)	200	200	200
Ірландія	12 (6+3+3)	183	min 183	167

Користувач задає назву країни. Вивести дані про навчання в цій країні.

Варіант № 29

Дані для занесення у файл:

Таблиця 1. Класифікація насіння основних зернових культур за масою 1000 н

Культура	Маса 1000 насінин, г			
	дуже дрібне (1 бал)	дрібне (2-3 бали)	середнє (4-5 балів)	крупне (6-7 балів)
Пшениця	<32	32-37	38-42	43-46
Жито	<30	30-33	34-36	37-40
Тритикале	<40	40-45	46-50	51-60
Ячмінь озимий	<35	35-40	41-45	46-50
Ячмінь ярий	<40	40-45	46-50	51-55
Овес	<30	30-35	35-40	41-45

Користувач задає назву культури. Вивести масу дрібного насіння.

Варіант № 30

Дані для занесення у файл:

Title	Votes per Year			
	2000	2001	2002	2003
★★★★ The Shawshank Redemption	40597	28949	37977	56514
★★★★ The Godfather	35153	23719	22352	15590
★★★★ The Godfather: Part II	24397	21372	17889	20143
★★★☆☆ Pulp Fiction	34680	22671	41965	18116
★★★☆☆ The Good, the Bad and the Ugly	13821	13642	17979	7400
★★★☆☆ The Dark Knight				

Користувач задає рік. Вивести назву фільму, який набрав більшість голосів у порівнянні з іншими роками.

Додаток К. Індивідуальні завдання до ЛР № 6

Таблиця К.1 – Завдання до ЛР № 6

<p>Варіант №1 Об'єм пам'яті – 150 байтів. Вивести адресу виділеної пам'яті. Вивести адреси розміщення кожного символу слова Laboratories.</p> <p><i>C is a general-purpose programming language created by Dennis Ritchie at the Bell Laboratories in 1972.</i></p>	<p>Варіант № 2 Об'єм пам'яті – 100 байтів. Вивести адресу виділеної пам'яті. Вивести вміст n-го та m-го байтів, n та m задається користувачем.</p> <p><i>C is strongly associated with UNIX, as it was developed to write the UNIX operating system.</i></p>	<p>Варіант № 3 Об'єм пам'яті – 90 байтів. Вивести адресу виділеної пам'яті. Вивести адресу слова, порядковий номер якого задається користувачем.</p> <p><i>If you know C, you will have no problem learning other popular programming languages.</i></p>	<p>Варіант № 4 Об'єм пам'яті – 85 байтів. Вивести адресу виділеної пам'яті. Вивести адреси першого та останнього слів.</p> <p><i>C is very versatile; it can be used in both applications and technologies.</i></p>
<p>Варіант № 5 Об'єм пам'яті – 125 байтів. Вивести адресу виділеної пам'яті. Вивести адреси всіх літер C, c.</p> <p><i>The main difference between C and C++ is that C++ support classes and objects, while C does not.</i></p>	<p>Варіант № 6 Об'єм пам'яті – 120 байтів. Вивести адресу виділеної пам'яті. Вивести адреси всіх входжень літери, заданої користувачем.</p> <p><i>An IDE (Integrated Development Environment) is used to edit and compile the code.</i></p>	<p>Варіант № 7 Об'єм пам'яті – 95 байтів. Вивести адресу виділеної пам'яті. Вивести адреси літер, якими закінчується кожне слово.</p> <p><i>To output values or print text in C, you can use the printf() function.</i></p>	<p>Варіант № 8 Об'єм пам'яті – 130 байтів. Вивести адресу виділеної пам'яті. Вивести адреси літер, якими починається кожне слово.</p> <p><i>The newline character (\n) is called an escape sequence, and it forces the cursor to change its position.</i></p>
<p>Варіант № 9 Об'єм пам'яті – 140 байтів. Вивести адресу виділеної пам'яті. Вивести адреси всіх входжень літер I, i.</p> <p><i>In C, there are different types of variables (defined with different keywords).</i></p>	<p>Варіант № 10 Об'єм пам'яті – 135 байтів. Вивести адресу виділеної пам'яті. Вивести адреси початку всіх слів.</p> <p><i>Explicit conversion is done manually by placing the type in parentheses in front of the value.</i></p>	<p>Варіант № 11 Об'єм пам'яті – 115 байтів. Вивести адресу виділеної пам'яті. Вивести адреси останньої літери всіх слів.</p> <p><i>The data type specifies the size and type of information the variable will store.</i></p>	<p>Варіант № 12 Об'єм пам'яті – 155 байтів. Вивести адресу виділеної пам'яті. Вивести адреси розміщення кожного символу слова automatically.</p> <p><i>Implicit conversion is done automatically by the compiler when you assign a value of one type to another.</i></p>

<p>Варіант № 13 Об'єм пам'яті – 160 байтів. Вивести адресу виділеної пам'яті. Вивести адреси другого символу слів want та keyword.</p> <p>If you don't want others (or yourself) to change existing variable values, you can use the const keyword.</p>	<p>Варіант № 14 Об'єм пам'яті – 160 байтів. Вивести адресу виділеної пам'яті. Вивести адреси всіх входжень слова if.</p> <p>Use the if statement to specify a block of code to be executed if a condition is true.</p>	<p>Варіант № 15 Об'єм пам'яті – 180 байтів. Вивести адресу виділеної пам'яті. Вивести адреси розміщення кожного символу слова variable.</p> <p>Arrays are used to store multiple values in a single variable, instead of separate variables for each value.</p>	<p>Варіант № 16 Об'єм пам'яті – 165 байтів. Вивести адресу виділеної пам'яті. Вивести вміст n-го та m-го байтів, n та m задається користувачем.</p> <p>Loops are handy because they save time, reduce errors, and they make code more readable.</p>
<p>Варіант № 17 Об'єм пам'яті – 190 байтів. Вивести адресу виділеної пам'яті. Вивести адресу слова, порядковий номер якого задається користувачем.</p> <p>Functions are used to perform certain actions, and they are important for reusing code.</p>	<p>Варіант № 18 Об'єм пам'яті – 175 байтів. Вивести адресу виділеної пам'яті. Вивести адреси першого та останнього слів.</p> <p>Parameters are specified after the function name, inside the parentheses.</p>	<p>Варіант № 19 Об'єм пам'яті – 125 байтів. Вивести адресу виділеної пам'яті. Вивести адреси всіх літер d.</p> <p>In C, you can create, open, read, and write to files by declaring a pointer of type FILE.</p>	<p>Варіант № 20 Об'єм пам'яті – 129 байтів. Вивести адресу виділеної пам'яті. Вивести адреси всіх входжень літери, заданої користувачем.</p> <p>Structures are a way to group several related variables into one place.</p>
<p>Варіант № 21 Об'єм пам'яті – 129 байтів. Вивести адресу виділеної пам'яті. Вивести адреси літер, якими закінчується кожне слово.</p> <p>For code optimization, it is recommended to separate the declaration and the definition of the function.</p>	<p>Варіант № 22 Об'єм пам'яті – 111 байтів. Вивести адресу виділеної пам'яті. Вивести адреси літер, якими починається кожне слово.</p> <p>C does not have a string type to easily create string variables.</p>	<p>Варіант № 23 Об'єм пам'яті – 140 байтів. Вивести адресу виділеної пам'яті. Вивести адреси всіх входжень літер f.</p> <p>To output the string, you can use the printf() function together with the format specifier %s.</p>	<p>Варіант № 24 Об'єм пам'яті – 133 байтів. Вивести адресу виділеної пам'яті. Вивести адреси початку всіх слів.</p> <p>Since strings are actually arrays in C, you can access a string by referring to its index number [].</p>
Варіант № 25	Варіант № 26	Варіант № 27	Варіант № 28

<p>Об'єм пам'яті – 113 байтів. Вивести адресу виділеної пам'яті. Вивести адреси останньої літери всіх слів.</p> <p>C has many useful string functions, which can be used to perform certain operations on strings.</p>	<p>Об'єм пам'яті – 155 байтів. Вивести адресу виділеної пам'яті. Вивести адреси розміщення кожного символу слова sizeof.</p> <p>It is important that you know that sizeof will always return the memory size (in bytes).</p>	<p>Об'єм пам'яті – 177 байтів. Вивести адресу виділеної пам'яті. Вивести адреси другого символу слів We та with.</p> <p>We can get the memory address of a variable with the reference operator &.</p>	<p>Об'єм пам'яті – 157 байтів. Вивести адресу виділеної пам'яті. Вивести адреси літер, якими закінчується кожне слово.</p> <p>A pointer is a variable that stores the memory address of another variable.</p>
<p>Варіант № 29 Об'єм пам'яті – 164 байти. Вивести адресу виділеної пам'яті. Вивести адреси всіх входжень слова the.</p> <p>The memory address of the first element is the same as the name of the array.</p>	<p>Варіант № 30 Об'єм пам'яті – 143 байти. Вивести адресу виділеної пам'яті. Вивести адреси всіх входжень літери, заданої користувачем.</p> <p>Comments can be used to explain code, and to make it more readable.</p>		

Додаток Л. Індивідуальні завдання до ЛР № 7

Таблиця Л.1 – Завдання до ЛР № 7

Варіант № 1 Таблиця В.4 – формула 12.	Варіант № 2 Таблиця В.4 – формула 15.
Варіант № 3 Таблиця В.4 – формула 16.	Варіант № 4 Таблиця В.4 – формула 17.
Варіант № 5 Таблиця В.4 – формула 14.	Варіант № 6 Таблиця В.4 – формула 19.
Варіант № 7 Таблиця В.4 – формула 20.	Варіант № 8 Таблиця В.3 – формула 12.
Варіант № 9 Таблиця В.5 – формула 1.	Варіант № 10 Таблиця В.5 – формула 2.
Варіант № 11 Таблиця В.5 – формула 3.	Варіант № 12 Таблиця В.5 – формула 4.
Варіант № 13 Таблиця В.5 – формула 5.	Варіант № 14 Таблиця В.5 – формула 6.
Варіант № 15 Таблиця В.5 – формула 7.	Варіант № 16 Таблиця В.5 – формула 8.
Варіант № 17 Таблиця В.5 – формула 9.	Варіант № 18 Таблиця В.5 – формула 10.
Варіант № 19 Таблиця В.5 – формула 11.	Варіант № 20 Таблиця В.5 – формула 12.
Варіант № 21 Таблиця В.5 – формула 13.	Варіант № 22 Таблиця В.5 – формула 14.
Варіант № 23 Таблиця В.5 – формула 15.	Варіант № 24 Таблиця В.6 – формула 3.
Варіант № 25 Таблиця В.6 – формула 4.	Варіант № 26 Таблиця В.6 – формула 8.
Варіант № 27 Таблиця В.6 – формула 9.	Варіант № 28 Таблиця В.6 – формула 11.
Варіант № 29 Таблиця В.6 – формула 12.	Варіант № 30 Таблиця В.6 – формула 15.

Додаток М. Індивідуальні завдання до ЛР № 8

Таблиця М.1 – Завдання до ЛР № 8

Варіант № 1 Нескінченний цикл: виведення слова Attention через кожні 2 сек. Обробка переривання: виведення 10-ти звукових сигналів.	Варіант № 2 Нескінченний цикл: виведення слова SIGNAL через кожні 3 сек. Обробка переривання: виведення 9-ти звукових сигналів.	Варіант № 3 Нескінченний цикл: виведення слова Interruption через кожні 4 сек. Обробка переривання: виведення звукового сигналу.	Варіант № 4 Нескінченний цикл: виведення слова Program через кожні 5 сек. Обробка переривання: виведення 3-х звукових сигналів.
Варіант № 5 Нескінченний цикл: виведення слова FUNCTION через кожні 6 сек. Обробка переривання: виведення 4-х звукових сигналів.	Варіант № 6 Нескінченний цикл: виведення слова Mathematics через кожні 10 сек. Обробка переривання: виведення 5-ти звукових сигналів.	Варіант № 7 Нескінченний цикл: виведення слова Language через кожні 7 сек. Обробка переривання: виведення 6-ти звукових сигналів.	Варіант № 8 Нескінченний цикл: виведення слова Types через кожні 8 сек. Обробка переривання: виведення 7-и звукових сигналів.
Варіант № 9 Нескінченний цикл: виведення слова Macros через кожні 9 сек. Обробка переривання: виведення 11-ти звукових сигналів.	Варіант № 10 Нескінченний цикл: виведення слова Integer через кожні 12 сек. Обробка переривання: виведення 3-х звукових сигналів.	Варіант № 11 Нескінченний цикл: виведення слова Header через кожні 11 сек. Обробка переривання: виведення 13-ти звукових сигналів.	Варіант № 12 Нескінченний цикл: виведення слова FILE через кожні 5 сек. Обробка переривання: виведення 14-ти звукових сигналів.
Варіант № 13 Нескінченний цикл: виведення слова Library через кожні 9 сек. Обробка переривання: виведення 15-ти звукових сигналів.	Варіант № 14 Нескінченний цикл: виведення слова IDE через кожні 4 сек. Обробка переривання: виведення 16-ти звукових сигналів.	Варіант № 15 Нескінченний цикл: виведення слова Compiler через кожні 16 сек. Обробка переривання: виведення 7-и звукових сигналів.	Варіант № 16 Нескінченний цикл: виведення слова Binary через кожні 4 сек. Обробка переривання: виведення 18-ти звукових сигналів.
Варіант № 17 Нескінченний цикл: виведення слова Float через кожні 10 сек. Обробка переривання: виведення 10-ти звукових сигналів.	Варіант № 18 Нескінченний цикл: виведення слова Array через кожні 5 сек. Обробка переривання: виведення 5-ти звукових сигналів.	Варіант № 19 Нескінченний цикл: виведення слова Structure через кожні 6 сек. Обробка переривання: виведення 6-ти звукових сигналів.	Варіант № 20 Нескінченний цикл: виведення слова Union через кожні 20 сек. Обробка переривання: виведення 2-х звукових сигналів.
Варіант № 21 Нескінченний цикл: виведення слова	Варіант № 22 Нескінченний цикл: виведення слова	Варіант № 23 Нескінченний цикл: виведення слова	Варіант № 24 Нескінченний цикл: виведення слова

Statement через кожні 11 сек. Обробка переривання: виведення 11-ти звукових сигналів.	Alignment через кожні 12 сек. Обробка переривання: виведення 12-ти звукових сигналів.	INPUT через кожні 8 сек. Обробка переривання: виведення 8-и звукових сигналів.	Environment через кожні 25 сек. Обробка переривання: виведення 5-ти звукових сигналів.
Варіант № 25 Нескінченний цикл: виведення слова Iteration через кожні 22 сек. Обробка переривання: виведення 6-ти звукових сигналів.	Варіант № 26 Нескінченний цикл: виведення слова Pointer через кожні 13 сек. Обробка переривання: виведення 13-ти звукових сигналів.	Варіант № 27 Нескінченний цикл: виведення слова Standart через кожні 2 сек. Обробка переривання: виведення 20-ти звукових сигналів.	Варіант № 28 Нескінченний цикл: виведення слова Paradigm через кожні 24 сек. Обробка переривання: виведення 9-ти звукових сигналів.
Варіант № 29 Нескінченний цикл: виведення слова Network через кожні 23 сек. Обробка переривання: виведення 11-ти звукових сигналів.	Варіант № 30 Нескінченний цикл: виведення слова Information через кожні 30 сек. Обробка переривання: виведення 30-ти звукових сигналів.		

ЛІТЕРАТУРА

1. ПРОГРАМУВАННЯ-2. Мова С [Електронний ресурс]: навч. посіб. для студ. спеціальності 113 «Прикладна математика», освітньої програми «Наука про дані (Data Science) та математичне моделювання» / КПІ ім. Ігоря Сікорського; уклад.: Н.Д. Любашенко. – Електронні текстові дані (1 файл: 1,6 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2019. – 144 с.
2. Програмування мовою С: методичні вказівки до виконання лабораторних робіт із кредитного модуля «Програмування-2. С» для студентів спеціальності «Прикладна математика» [Текст] / В. В. Громова. — К. : НТУУ «КПІ», 2016. — 193 с.
3. Мова програмування С. Модульний підхід : навчальний посібник / В.Г. Акуловський, В.О. Яковенко, Ю.В. Ульяновська, В.В. Костенко, Д.Є. Костенко ; Міністерство освіти і науки України, Університет митної справи та фінансів. – Дніпро : Університет митної справи та фінансів, 2018. – 217 с.
4. ISO/IEC 9899:2018 Information technology – Programming languages – C. <https://www.open-std.org/jtc1/sc22/wg14/www/docs/n2596.pdf>.

Навчальне видання

**Любашенко Наталія Дмитрівна
Костюшко Ірина Анатоліївна**

**Програмування.
Частина 2.
Вказівки до виконання
лабораторних робіт**

Навчальний посібник

Міністерство освіти та науки України

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Свідоцтво про державну реєстрацію: серія ДК № 5354 від 25.05.2017 р.
просп. Перемоги, 37,
м. Київ, 03056

Електронне мережне навчальне видання

Київ
КПІ ім. Ігоря Сікорського
2023