

Практичне заняття 02

Алгоритмічна основа програмування

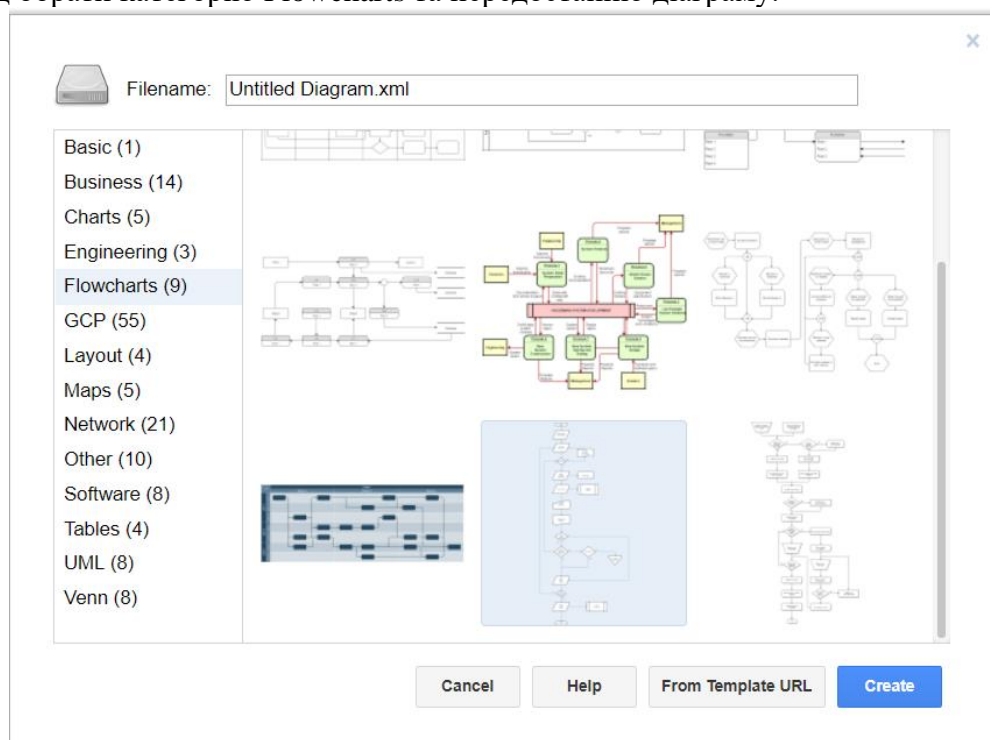
Система оцінювання

№	Тема	К-ть балів
1.	<i>Побудова блок-схем</i>	5*
2.	<i>Базовий ввід-вивід мовою програмування C</i>	2*
3.	<i>Застосування основних операторів</i>	4*
4.	<i>Використання управляючих інструкцій</i>	4,5*
5.	<i>Здача звіту</i>	0,5
	Всього	16

* – діє бонусна система




1. Побудова блок-схем

Для побудови блок-схеми можна використовувати сервіс www.draw.io. При створенні діаграми слід обрати категорію Flowcharts та передостанню діаграму.

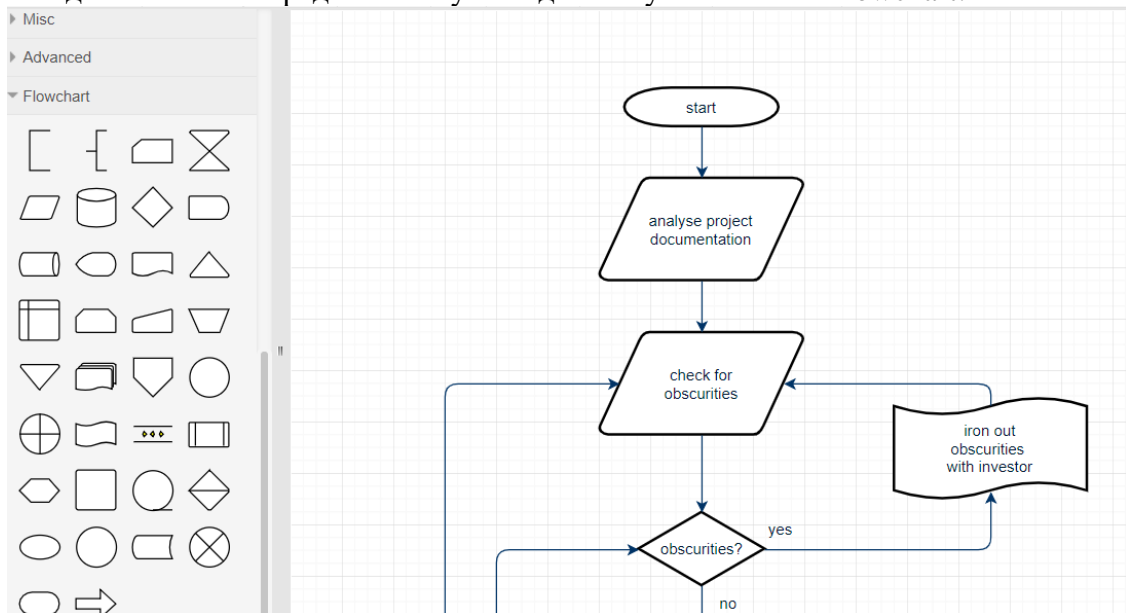


Елементи діаграми, що потрібні для побудови блок-схеми, знаходяться у випадаючому меню Flowchart. Блок-схема має наступні елементи:

	Термінатор – відображає вхід у зовнішнє середовище (початок) або вихід з нього (кінець)
	Процес – відображає одну або кілька операцій, обробку даних будь-якого виду
	Дані - відображає перетворення у форму, придатну для обробки (введення) або відображення результатів обробки (виведення).

	Рішення – відображає обробку умови, рішення або функцію перемикального типу з одним входом і двома або більше альтернативними виходами, з яких тільки один може бути обраний після обчислення умов, визначених всередині цього елемента.
	Цикл з параметром – відображає заголовок циклу з параметром. У ньому через крапку з комою вказуються ім'я змінної (параметра) з початковим значенням, граничне значення параметра (або умова виконання циклу), крок зміни параметра.
	Зумовлений процес - відображає виконання процесу, що складається з однієї або кількох операцій, що визначені в іншому місці програми (у підпрограмі, модулі).

Наведені елементи представлені у випадаючому зліва меню Flowchart:



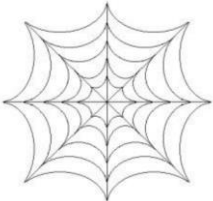
Для виконання практичного заняття студент виконує загальні завдання та завдання відповідно до свого варіанту (таблиця 2). Передбачені завданнями алгоритми опишіть у словесному вигляді, наближеному до псевдокоду, та у вигляді блок-схеми. При вирішенні поставлених задач спочатку слід запропонувати перелік вхідних даних та вигляд результату роботи алгоритму.

1. Алгоритм знаходження максимуму в наборі значень передбачає наступні кроки: встановлення в якості максимуму першого числа з набору та подальше порівняння максимуму з рештою чисел. Якщо зустрілось число, що буде більше за поточний максимум, дане значення стає максимальним. Аналогічно відбувається порівняння для знаходження мінімуму. Опишіть алгоритм, який буде обчислювати в один прохід максимум та мінімум для набору чисел.
2. [Шифр Скитала](#) – найстаріший шифр перестановки, який змінює порядок букв у тексті шляхом намотування стрічки з текстом на палицю відповідної товщини. Опишіть алгоритм шифрування та дешифрування тексту за допомогою палиці.



3. Запропонуйте алгоритм виходу з лабіринту та опишіть його.
4. Опишіть алгоритм відповідно до свого варіанту. Опис повинен містити джерело інформації (посилання), звідки були взяті ідеї для побудови алгоритму.

Таблиця 2. Завдання відповідно до варіанту (номеру в списку підгрупи)

№ у списку	Алгоритм	
1	Опишіть алгоритм купівлі товару в онлайн-магазині.	
2	Опишіть свій алгоритм прибуття в бізнес-коледж.	
3	Дивлячись на рисунок, опишіть покроково процедуру його прорисовки. Зауважте, що зображення характерна симетричність, тобто можна рисувати частини зображення за допомогою однієї процедури з різними параметрами.	
4	Сплануйте походи до магазину на тиждень, знаючи, яких продуктів та на скільки вистачатиме	
5	Запропонуйте та опишіть алгоритм підбору пари шкарпеток з купи	
6	Опишіть алгоритм роботи кавового автомата. Про його роботу можете глянути тут .	
7	Опишіть алгоритм карткової гри «Дурень»	
8	Опишіть алгоритм черговості зв'язку з людиною (наприклад, телефон, зустріч, месенджери та ін.)	
9	Опишіть свій алгоритм прибуття в бізнес-коледж	
10	Опишіть процес приготування страви відповідно до рецепту, який можна обрати за адресою https://rud.ua/consumer/recipe/	
11	Опишіть алгоритм генерального прибирання квартири	
12	Опишіть алгоритм збирання автомобіля на конвеєрі. За основу можна, наприклад, взяти статтю https://www.the-village.ru/village/business/process/171607-kak-delayut-mashiny	
13	Опишіть алгоритм вирощування рослин. Для прикладу можна взяти посилання https://www.babyblog.ru/community/post/baby4833/3153575 або подібні джерела.	

2. Базовий ввід-вивід мовою програмування C

Крок 1. Основні можливості для вводу інформації в програму та виводу даних з неї постачаються в заголовковому файлі `stdio.h` (standard input-output – стандартний ввід-вивід). Вміст цього файлу компілюється разом з Вашим кодом, якщо на початку цього коду буде включено інструкцію

```
#include <stdio.h>
```

Зчитування та запис інформації може відбуватись по-різному: окремими символами, рядками, файлами тощо. На початку всі працюючі програми мовою C у своєму коді містять функцію `main()`, що є точкою входу в програму – місцем, звідки програма почне виконуватись інструкція за інструкцією.

```
#include <stdio.h>
int main()
{
    // Display text
    printf("Hello, world!");
    return 0;
}
```

У даному тексті всередині функції `main()` присутні наступні інструкції:

- 1) Коментар `«// Display text»`, який дозволяє робити розробнику програми помітки з поясненнями коду. Вони ігноруються компілятором, проте допомагають зрозуміти написаний код іншим людям. Короткий коментар записується одним рядком так: `// текст коментаря`

Більш розлогий коментар, що складається з кількох рядків, записується наступним чином:

```
/*Це
довгий
коментар
*/
```

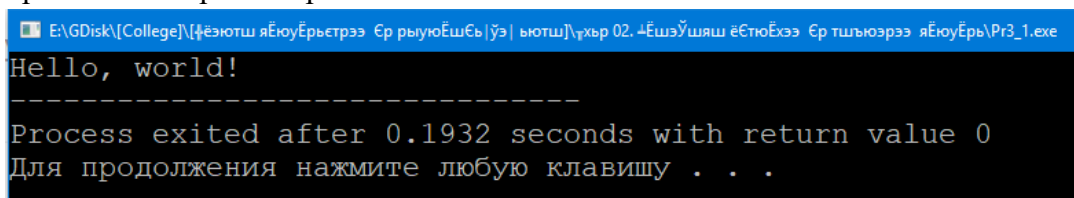
- 2) Команда на вивід інформації в консоль:

```
printf("Hello, world!");
```

складається з виклику функції `printf()`, яка реалізована всередині `stdio.h`. Функція приймає аргументи, у даному випадку – це текстовий рядок `«Hello, world!»`. Інструкція закінчується оператором `«;»`.

- 3) Оператор `return` виконує повернення результатів роботи функції. Вигляд повернення залежить від функції, тут – це код помилки, яка може трапитись під час її роботи. Код `«0»` означає, що проблем немає, проте за бажанням можна записати інші коди, що вказуватимуть на наявність проблем.

Вигляд виводу показано на рис. 1. Слід зауважити, що часто можуть бути проблеми з відображенням кириличних символів. Наприклад, на рис. 1 видно, що кирилиця в шляху до файлу відображається «крякозябрами».



```
E:\GDisk\College\Цезютш яЁюёрьстрээ Ёр рыуюёшёь|ўэ| ьютш|ўхър 02.4ёшэўшыш ёётюёхээ Ёр тшьюэрээ яЁюёрь\Pr3_1.exe
Hello, world!
-----
Process exited after 0.1932 seconds with return value 0
Для продолжения нажмите любую клавишу . . .
```

Рис. 1. Вивід першої програми

У заголовковому файлі `locale.h` присутня функція [`setlocale\(\)`](#), яка дозволяє частково виправити кодування:

```

1  #include <stdio.h>
2  #include <locale.h>
3
4  int main()
5  {
6      setlocale(LC_ALL, "");
7      // Виводить текст у консолі
8      printf("Привіт, світ!");
9      return 0;
10 }
```

```

E:\GDisk\College\{фёютш яёуёрьстрээ ёр рыуюёшёь|ўэ| ьотш\ухьр 02. 4ёшзўшыш ёётюёхээ ёр тшьюэрээ яёуёрь\Pr3_1.exe
Прив?т, св?т!
-----
Process exited after 0.03339 seconds with return value 0
Для продолжения нажмите любую клавишу . . .

```

Рис. 2. Вивід програми з частковою підтримкою кирилиці

Детальніше про проблеми роботи з кирилицею в мові програмування C можна почитати [тут](#).

Також функція `printf()` дозволяє виводити значення змінних та виразів у консоль. З цією метою слід використовувати так звані специфікатори формату, які задаватимуть тип значення, яке слід вивести. Серед найбільш поширених:

- `%d` – цілі десяткові числа (**d**ecimal),
- `%lf` – дробові числа подвійної точності (тип `double`, `lf` = **l**ong **f**loat)
- `%c` – символ (тип `char`)
- `%s` – рядок символів (**s**tring)

Інші специфікатори можна переглянути [тут](#). У прикладі 3 показано використання специфікаторів та продемонстровано результати їх роботи

```

1  #include <stdio.h>
2  #include <locale.h>
3
4  int main()
5  {
6      int age = 20;
7      double height = 1.69;
8      setlocale(LC_ALL, "");
9      // Виводить текст у консолі
10     printf("Привіт, світ! Мені %d років. Мій зріст - %lf м.", age, height);
11     return 0;
12 }
```

```

E:\GDisk\College\{фёютш яёуёрьстрээ ёр рыуюёшёь|ўэ| ьотш\ухьр 02. 4ёшзўшыш ёётюёхээ ёр тшьюэрээ яёуёрь\Pr3_1.exe
Прив?т, св?т! Мен? 20 рок?в. М?й зр?ст - 1,690000 м.
-----
Process exited after 0.04252 seconds with return value 0
Для продолжения нажмите любую клавишу . . .

```

Рис. 3. Вивід тексту з використанням специфікаторів формату

Спробуйте вивести свою коротку автобіографію в консолі та додайте код і скриншот запуску до загального звіту.	Створіть програму, що показує Ваше ім'я та повну поштову адресу в форматі, характерному для листів Укрпошти. Вашій програмі не потрібно зчитувати нічого від користувача.
---	---

Крок 2. (Форматований вивід) Розширений огляд можливостей функції printf() доступний [тут](#). Зокрема, для виконання вирівнювання значень по правому краю буде корисним налаштування ширини специфікатора (width option). Воно дозволяє «забронювати» визначену кількість знакомісць під представлення тексту, вивести незначущі нулі, знаки + та -. Для цього між у записі специфікатора між знаком «%» та відповідними символами ставиться числове представлення кількості цих знакомісць, наприклад %-5.3lf (дробове число з 5ма знаками для цілої та трьома знаками для дробової частин, вирівняне з лівого боку). Спробуйте вивести на екран видаткову накладну, представлену на рис. 4.

Крок 3. (Форматований ввід даних). Для вводу даних частіше всього використовується функція scanf(), детальнішу інформацію про яку можна знайти, наприклад, [тут](#). Напишіть код, який зчитуватиме значення з основними специфікаторами.

Видаткова накладна № 25 від 29.05.2018 р.					
Постачальник:	ТОВ «Едельвейс», р/р 260059999999999 у банку АКБ «Азаль», м. Чернігів, МФО 300335 14000, м. Чернігів, пр. Перемоги, 153, тел. +380(46)151554				
Покупець:	ТОВ «Орхідеа», р/р 260053333333333 у банку АКБ «Приватбанк», м. Чернігів, МФО 353586 14032, м. Чернігів, вул. Белова, 37, тел. +380(46)181864				
Договір:	купівлі-продажу № 12 від 15.05.2018 р.				
№	Товар	Кіль-ть	Од.	Ціна без ПДВ	Сума без ПДВ
1	лак меблевий акрил-поліуретановий Тгае Lux Moebel lak (0,25л)	10	банки	150,00	1500,00
2	ДВП СТ-40 (2.5мм×2440 мм ×1220 мм)	5	лист	95,00	475,00
Всього найменувань 2.				Всього:	1975,00
				Сума без ПДВ:	395,00
				Всього з ПДВ:	2370,00
Дві тисячі триста сімдесят гривень 00 копійок, у т.ч. ПДВ: Триста дев'яносто п'ять гривень 00 копійок					
Від постачальника:	Садчиков	Садчиков В.С.	Отримав(ла)	Дубина	Дубина М.В.
			За довіреністю №		
			5 від 29.05.2018 р.		

Рис. 4. Видаткова накладна для виводу в консоль

Крок 4 (Символьний ввід-вивід). Напишіть програму, яка дозволить ввести текст та для кожного символу з нього виведе його код у таблиці ASCII. З цією метою можна використовувати функції getchar() та putchar().

3. Застосування основних операторів

Оператори мови програмування C поділяються на кілька категорій:

- Арифметичні оператори (+, -, *, /, %, ++, --);
- Оператори відношення (==, !=, >, <, >=, <=);
- Логічні оператори (&&, ||, !);
- Побітові оператори (&, |, ^, ~, <<, >>);
- Оператори присвоєння (=, +=, -=, *=, /=, %=, <<=, >>=, &=, ^=, |=);
- Інші оператори (sizeof(), & (взяття адреси), * (вказівник), ?:).

Пріоритетність операторів можна знайти, наприклад, [тут](#). Реалізуйте мовою програмування C наступні задачі:

1. ^{0,2 бала} Програма в цій вправі повинна починати роботу зі зчитування вартості їжі, замовленої з ресторану (назва страви, кількість порцій, ціна за порцію). Потім програма обчислить податок та чайові за їжу. Розраховуйте на чайові в розмірі 14% від замовлення та податок у розмірі 18% від замовлення. Вивід вашої програми має включати розмір податку, чайових та загальну суму для оплати. Формат виводу передбачає представлення всіх значень з використанням 2 знаків після коми.
2. ^{0,2 бала} Напишіть програму, яка починає зі зчитування радіусу *r* від користувача. Програма продовжуватиме роботу, обчислюючи та виводячи на екран площу круга з радіусом *r* та об'єм кулі з радіусом *r*. Використовуйте константу *M_PI* з заголовкового файлу *math.h*.
3. ^{0,2 бала} Створіть змінні-вказівники на значення різних типів (*int*, *float*, *double*, *char* та *in.*), а також *NULL*-вказівник. За допомогою оператора *sizeof()* виведіть їх значення, розмір у байтах, а також розмір самих типів, що їм відповідають. Оформіть результат у вигляді таблиці з вирівнюванням стовпців. Визначте за адресами, чи в пам'яті дані змінні були алоковані послідовно одна за одною?
4. ^{0,2 бала} Напишіть програму, яка починає роботу зі зчитування кількості секунд від користувача. Далі вона має вивести еквівалентний проміжок часу у формі Д:ГГ:ХХ:СС (дні, години, хвилини, секунди). Потрібно, щоб форматований вивід передбачав по 2 знакових місця для годин, хвилин та секунд з дописуванням незначущого нуля за потреби.
5. ^{0,2 бала} Для заданого RGB-кольору, представленого у вигляді шістнадцяткового числа, яке вводить користувач, визначте окремі значення для червоного, зеленого та синього каналів. Для цього слід використати побітові оператори.

4. Використання управляючих інструкцій

1. ^{0.1 бала} У деякій компанії робітників оцінюють у кінці року. Шкала оцінювання починається з 0.0, більше значення означає кращу продуктивність праці та, відповідно, більший премії. Оцінка для премії може бути 0.0, 0.4 або понад 0.6. Значення між 0.0 і 0.4, а також між 0.4 і 0.6 ніколи не використовуються. Смысл коефіцієнтів описано в таблиці. Розмір премії складає 4173 грн. * коефіцієнт.

Rating	Meaning
0.0	Unacceptable performance
0.4	Acceptable performance
0.6 or more	Meritorious performance

Напишіть програму, яка зчитує рейтинг робітника та характеризує його роботу відповідно до таблиці. Також програма має повідомити про розмір премії для робітника. Від'ємні значення програма має вважати помилковими.

2. ^{0.1 бала} З 1 липня 2018 року в Україні діє правило заокруглення готівкових розрахунків за умови відсутності монет відповідного номіналу в продавця чи покупця. Напишіть програму, яка зчитує ціни від користувача, поки не буде введено порожній рядок. Додаток повинен обчислити суму оплати в готівковій (округлити до найближчого 10-копієчника) та безготівковій формі. Одним із способів округлення є такий алгоритм дій:
 - 1) Визначити кількість копійок у сумарній вартості
 - 2) Обчислити остачу від ділення на 10.
 - 3) Округлити до меншого, якщо остача від ділення менша за 5. Інакше округлити до більшого.
3. ^{0.1 бала} Вас найняли в NASA, щоб розробити програму, яка пропонує ввести користувачеві свою масу, а потім обрати планету. Додаток має обчислити та вивести Вашу масу на обраній планеті з наступними коефіцієнтами:
 - 1) Меркурій - 0.38
 - 2) Венера - 0.91
 - 3) Земля - 1.0
 - 4) Марс - 0.38
 - 5) Юпітер - 2.34
 - 6) Сатурн - 1.06
 - 7) Уран - 0.92
 - 8) Нептун - 1.19
4. ^{0.2 бала} Напишіть програму, яка конвертує десяткове число в двійкове. Зчитуйте десяткове число від користувача як ціле число, а потім використовуйте алгоритм ділення для виконання перетворення. Коли алгоритм завершує роботу, змінна `result` міститиме двійкове представлення числа. Відобразіть результат на екрані з відповідним пояснювальним текстом.
5. ^{0.1 бала} Виведіть на екран табличку множення.
6. ^{0.2 бала} Андроїд Вася успішно здав усі екзамени і тепер хоче дізнатись, чи буде в нього стипендія. В університеті наступна політика її надання:
 - a. Якщо у студента є трійки, стипендія йому не виплачується;
 - b. Якщо студент здав сесію лише на п'ятірки, він отримує іменну стипендію.
 - c. Якщо студент не отримав іменну стипендію, але його середній бал дорівнює або перевищує 4.5, він отримує підвищену стипендію.
 - d. Інакше студент отримує звичайну стипендію.

Напишіть програму, яка допоможе визначити, чи є стипендія у Васі, і якщо так, то яка саме. На вхід подаються оцінки за сесію, а на виході виводиться тип стипендії.

Наприклад:

исходные данные	результат
3 5 5 4	High
3 3 3 3	None

Проаналізуйте структуру запропонованої програми та запишіть відповідні лексеми (токени) до таблиці згідно з варіантом. Для оператора вкажіть у дужках його пріоритет.

№	Тип лексеми	Лексема	Рядки, в яких зустрічається
1.	Ключове слово		
2.	Ідентифікатор		
3.	Константа		
4.	Рядок		
5.	Спеціальний символ		
6.	Оператор		

Ключові слова в мові C:

<i>auto</i>	<i>double</i>	<i>int</i>	<i>struct</i>
<i>break</i>	<i>else</i>	<i>long</i>	<i>switch</i>
<i>case</i>	<i>enum</i>	<i>register</i>	<i>typedef</i>
<i>char</i>	<i>extern</i>	<i>return</i>	<i>union</i>
<i>const</i>	<i>short</i>	<i>float</i>	<i>unsigned</i>
<i>continue</i>	<i>for</i>	<i>signed</i>	<i>void</i>
<i>default</i>	<i>goto</i>	<i>sizeof</i>	<i>volatile</i>
<i>do</i>	<i>if</i>	<i>static</i>	<i>while</i>

Спеціальні символи в мові C:

, (comma)	{ (opening curly bracket)
. (period)	} (closing curly bracket)
; (semi-colon)	[(left bracket)
: (colon)] (right bracket)
? (question mark)	((opening left parenthesis)
' (apostrophe)) (closing right parenthesis)
" (double quotation mark)	& (ampersand)
! (exclamation mark)	^ (caret)
(vertical bar)	+ (addition)

/ (forward slash)	- (subtraction)
\ (backward slash)	* (multiplication)
~ (tilde)	/ (division)
_ (underscore)	> (greater than or closing angle bracket)
\$ (dollar sign)	< (less than or opening angle bracket)
% (percentage sign)	# (hash sign)

Оператори в мові C з відповідними пріоритетами:

Пріоритет	Оператор	Опис	Асоціативність
1	++ --	Суфіксий/постфіксий інкремент та декремент	Зліва направо
	()	Виклик функції	
	[]	Звернення до елементу масиву	
	.	Доступ до члену структури чи об'єднання	
	->	Доступ до члена структури чи об'єднання за вказівником	
	(тип) { список }	Складений літерал (C99)	
2	++ --	Префіксий інкремент та декремент ^[1]	Справа наліво
	+ -	Унарний плюс або мінус	
	! ~	Логічне НЕ та побітове НЕ	
	(type)	Зведення типу	
	*	Розіменування (indirection, dereference)	
	&	Взяття адреси	
	sizeof	Отримання розміру ^[2]	
	_Alignof	Вирівнювання (C11)	
3	* / %	Множення, ділення та остача	Зліва направо
4	+ -	Додавання та віднімання	
5	<< >>	Bitwise left shift and right shift	
6	< <=	For relational operators < and ≤ respectively	
	> >=	For relational operators > and ≥ respectively	
7	== !=	For relational = and ≠ respectively	
8	&	Bitwise AND	
9	^	Bitwise XOR (exclusive or)	
10		Bitwise OR (inclusive or)	
11	&&	Logical AND	
12		Logical OR	
13	?:	Тернарний умовний оператор	Справа наліво
14 ^[3]	=	Simple assignment	
	+= -=	Assignment by sum and difference	
	*= /= %=	Assignment by product, quotient, and remainder	
	<<= >>=	Assignment by bitwise left shift and right shift	
	&= ^= =	Assignment by bitwise AND, XOR, and OR	
15	,	Comma	Зліва направо

[1] The operand of prefix ++ and -- can't be a type cast. This rule grammatically forbids some expressions that would be semantically invalid anyway. Some compilers ignore this rule and detect the invalidity semantically.

[2] The operand of sizeof can't be a type cast: the expression sizeof (int) * p is unambiguously interpreted as (sizeof(int)) * p, but not sizeof((int)*p).

[3] Assignment operators' left operands must be unary (level-2 non-cast) expressions. This rule grammatically forbids some expressions that would be semantically invalid anyway. Many compilers ignore this rule and detect the invalidity semantically. For example, e = a < d ? a++ : a = d is an expression that cannot be parsed because of this rule. However, many compilers ignore this rule and parse it as e = (((a < d) ? (a++) : a) = d), and then give an error because it is semantically invalid.

Варіант 1

```
1  /*****
2  C Program to calculate frequency of vowels in a string
3  *****/
4  #include <stdio.h>
5  #include <conio.h>
6
7  void main() {
8      int a = 0, e = 0, i = 0, o = 0, u = 0, sum = 0;
9      char c;
10
11     printf("Enter string: ");
12     printf("\nString will be terminated if you press Ctrl-Z & then
13 ENTER.");
14     printf("\nSTRING: ");
15     while ((c=getchar()) != EOF) {
16         if (c=='a' || c=='A')          a=a+1;
17         if (c=='e' || c=='E')          e=e+1;
18         if (c=='i' || c=='I')          i=i+1;
19         if (c=='o' || c=='O')          o=o+1;
20         if (c=='u' || c=='U')          u=u+1;
21     }
22     sum = a + e + i + o + u;
23
24     printf("\nFrequency of vowel 'a' is %d.", a);
25     printf("\nFrequency of vowel 'e' is %d.", e);
26     printf("\nFrequency of vowel 'i' is %d.", i);
27     printf("\nFrequency of vowel 'o' is %d.", o);
28     printf("\nFrequency of vowel 'u' is %d.", u);
29     printf("\nTotal no. of vowels in the text is %d.", sum);
30 }
```

Варіант 2

```
1  /*****
2  C - PALINDROMIC NUMBERS
3  *****/
4  #include <stdio.h>
5
6  int main() {
7      int n, reverse = 0, temp;
8
9      printf("Enter a number to check if it is a palindrome or not\n");
10     scanf("%d", &n);
11 }
```

```

12     temp = n;
13
14     while( temp != 0 ) {
15         reverse = reverse * 10;
16         reverse = reverse + temp%10;
17         temp = temp/10;
18     }
19
20     if ( n == reverse )
21         printf("%d is a palindrome number.\n", n);
22     else
23         printf("%d is not a palindrome number.\n", n);
24
25     return 0;
26 }

```

Варіант 3

```

1  /**
2  C PROGRAM TO CHECK GIVEN NUMBER IS ARMSTRONG NUMBER OR NOT
3  */
4  #include <stdio.h>
5
6  int Power(int n, int r) {
7      int c, p = 1;
8
9      for (c = 1; c <= r; c++)
10         p = p*n;
11
12     return p;
13 }
14
15 int main() {
16     int n, sum = 0, temp, remainder, digits = 0;
17
18     printf("Input an integer : \n");
19     scanf("%d", &n);
20
21     temp = n;
22     // Count number of digits
23     while (temp != 0) {
24         digits++;
25         temp = temp/10;
26     }
27
28     temp = n;
29
30     while (temp != 0) {
31         remainder = temp%10;
32         sum = sum + Power(remainder, digits);
33         temp = temp/10;
34     }
35
36     if (n == sum)
37         printf("%d is an Armstrong number.\n", n);
38     else
39         printf("%d is not an Armstrong number.\n", n);
40
41     return 0;
42 }

```

Варіант 4

```
1  /**
2  Bubble Sort Algorithm C Example
3  */
4  #include <stdio.h>
5  #include <stdlib.h>
6
7  void PrintArray(int *array, int n) {
8      for (int i = 0; i < n; ++i)
9          printf("%d ", array[i]);
10     printf("\n");
11 }
12
13 void BubbleSort(int array[], int n) {
14     bool swapped = true;
15     int j = 0;
16     int temp;
17
18     while (swapped) {
19         swapped = false;
20         j++;
21         for (int i = 0; i < n - j; ++i) {
22             if (array[i] > array[i + 1]) {
23                 temp = array[i];
24                 array[i] = array[i + 1];
25                 array[i + 1] = temp;
26                 swapped = true;
27             }
28         }
29     }
30 }
31
32 int main() {
33     int array[] = {94, 42, 50, 95, 333, 65, 54, 456, 1, 1234};
34     int n = sizeof(array)/sizeof(array[0]);
35
36     printf("Before Bubble Sort :\n");
37     PrintArray(array, n);
38
39     BubbleSort(array, n);
40
41     printf("After Bubble Sort :\n");
42     PrintArray(array, n);
43     return (0);
44 }
```