

Лабораторна робота № 2

ВНУТРІШНЄ ПРЕДСТАВЛЕННЯ

ЦІЛОЧИСЕЛЬНИХ ДАНИХ

Мета заняття: ознайомитися з числовими типами даних, їх видами та особливостями; розглянути алгоритм внутрішнього (машинного) представлення чисел у відповідності з діапазоном в знакових і беззнакових форматах типів ShortInt (signed char), Byte (unsigned char), Integer (int), Word (unsigned int); написати програму на мові Ассемблер для опису чисел, перевірити правильність виконаних розрахунків.

Теоретичні відомості

Поняття типу даних. Числові типи даних.

Цілочисельні дані

Тип даних – характеристика, яку явно чи неявно надано об'єкту (змінній, функції, полю запису, константі, масиву тощо). Тип даних визначає множину припустимих значень, формат їхнього збереження, розмір виділеної пам'яті та набір операцій, які можна робити над даними.

Машинні типи даних. У всіх комп'ютерах, заснованих на цифровій електроніці, інформація на найнижчому рівні представляється у вигляді бітів (із значенням 0 або 1). Найменша адресована одиниця інформації називається байт (зазвичай як октет, який містить 8 бітів). Одиниця інформації, яка оброблюється інструкціями машинного коду, називається словом (станом на 2006 рік, зазвичай по 32 або 64 біти). Через наявність доповнюючого коду, машині не потрібно розрізняти знакове та беззнакове числа для більшості випадків.

Існує спеціальний набір арифметичних інструкцій, які використовують різні представлення бітів у слова, для операцій з рухомою крапкою.

Прості типи даних. Мови програмування представляють деякі прості типи даних (або примітивні), як базові блоки для програм та спеціалізованіших складних типів даних. Зазвичай прості типи даних включають числові (кілька цілих та дійсних типів), логічний (булевий), символний та байтовий.

Існують такі числові типи даних:

- цілі та дійсні;
- беззнакові та знакові;

Цілі числа (англ. integer) не можуть містити у собі дріб. Для від'ємного числа треба ставити знак мінус (-) перед значенням (числом). Не можна використовувати кому у введенні такого числа, бо інакше буде викликана синтаксична помилка. Приклади цілих чисел:

42
10000
-233000
-100

Дійсні числа – це числа, які можуть містити у собі як цілі, так і дробові значення з точкою відокремлення від цілої частини. Для від'ємного числа треба ставити знак мінус (-) перед значенням (числом). Приклади дійсних чисел:

20.0005
99.9
-5000.12
-9999.9991

Кожне числовий тип даних має мінімальне та максимальне значення, яке називають діапазон значень. Важливо знати діапазон значень, особливо, коли працюєш з «маленькими» типами даних, оскільки у них можна зберігати лише значення у вузькому діапазоні. Спроба внести число, більше за доступний діапазон може призвести до помилок періоду компіляції/виконання, або до неправильних підрахунків (через відкидання) залежно від мови програмування, яка використовується.

Базові цілочислені типи даних для мови C та їх діапазони допустимих значень наведені в таблиці 2.1.

Таблиця 2.1

Цілочисельні типи даних

Тип	Діапазон значень	Потрібна пам'ять
Byte	0 ... 255	1 байт
Shortint	-128 ... 127	1 байт
Word	0 ... 65535	2 байти
Integer	-32768 ... 32767	2 байти
Longint	-2147483648 ... 2147483647	4 байти

Логічний тип даних – це тип даних, об'єкти якого можуть приймати одне з двох значень: істина (англ. true) та хибність (англ. false).

Перелічуваний тип, перелік (англ. enumeration type) – тип даних, що описується шляхом перелічення всіх можливих значень (кожне з яких позначається власним ідентифікатором), які можуть приймати об'єкти даного типу. Приклад (Pascal):

type Cardsuit = (clubs, diamonds, hearts, spades);

Символьний тип даних – це тип даних, що описує літери та інші знаки, використовувані на письмі. В залежності від мови програмування та конкретної реалізації, може займати 1 чи 2 байти, рідше 4. Однобайтний символьний тип може використовуватись для представлення символів з набору ASCII та інших восьмирозрядних кодувань, тоді як для представлення символів з набору Unicode потрібно щонайменше 2 байти.

До **байтових типів даних** відносяться:

- байт або byte (8 біт);
- слово або word (16 біт);
- подвійне слово або dword (32 біти);
- зчетверене слово або qword (64 біти);
- 10 байт ten byte (80 біт) і т.д.

Бітом називають двійкову цифру, одиничне значення 0 або 1. Групу з 8-ми бітів називають байтом. Байт заслужив своє власне ім'я з декількох причин. Елементарний елемент пам'яті (комірка) має довжину 8 бітів.

Упродовж кожного звертання до пам'яті процесор опрацьовує рівно 8 бітів інформації. **Байт** – це найменша одиниця інформації, з якою можна маніпулювати безпосередньо. Крім того, байт використовують для зображення одного символу. За допомогою одного байта можна визначити 256 (2⁸) окремих символів (межі для цілих чисел: -128 ... 127).

Для розміщення певного значення в одному байті пам'яті асемблер має в своєму розпорядженні спеціальний механізм визначення байта (define byte) – псевдокоманду (директиву) DB формату.

Отже, псевдокоманда DB:

- формує однобайтову константу - число;
- формує у пам'яті заданий рядок символів (літерал);
- просто резервує 1 байт пам'яті (за умови наявності).

Розмір **слова** становить **16 бітів**. Цей розмір визначено каналами передачі даних у процесорі. Над числами до 16-ти бітів операції можуть здійснюватися однією командою.

Псевдокоманда (директива) DW визначає слова (define word).

Залишився ще один тип даних, який постійно використовують у програмах мовою асемблера. Це – **подвійне слово**, значення у **32 біти** довжиною. У програмах використовують подвійні слова для зберігання адрес і дуже великих чисел. Щоб визначити область, яка містить значення подвійного слова, використовують псевдокоманду (директиву) DD (define doubleword). Ця псевдокоманда генерує поле розміром у 4 байти. Так само, як у випадку з DW, асемблер розміщує в пам'яті молодший байт подвійного слова нижче, а старший – вище. У такому ж порядку зберігаються середні два байти.

Подібно визначається пам'ять для **зчетвереного слова** (DQ) та поля у **10 байтів** (DT).

Складні типи даних – це типи, які складаються з елементів, що відносяться до простих типів. До складних типів даних відносяться: масиви; множини; рядки; записи; файли; динамічні змінні; вказівники; лінійні списки (стеки, черги); нелінійні списки (двійкові

дерева, несиметричні дерева, тексти, графи); процедурний тип; об'єкти.

Основні типи даних в Асемблері наведено в таблиці 2.2.

Таблиця 2.2

Основні типи даних в Асемблері

Тип	Директива	Кількість байт
Байт (byte)	DB	1
Слово (word)	DW	2
Подвійне слово (dword)	DD	4
Зчетверене слово (qword)	DQ	8
10 байт (tbyte)	DT	10

Алгоритм внутрішнього представлення цілочисельних даних

Внутрішнє представлення цілочисельних даних відбувається в два кроки. Спочатку задане число переводиться з десяткової системи числення в двійкову.

Для того, щоб змінити знак числа, потрібно інвертувати всі його біти і додати до нього одиницю – отримаємо представлення ВІД'ЄМНОГО числа в додатковому коді. Для цілих типів із знаком, під знак відводиться старший біт, причому для додатніх чисел він дорівнює 0, а для від'ємних – 1.

Потім отриману двійкову послідовність необхідно перевести в шістнадцяткову систему числення.

Приклад 1: Для заданих чисел ± 21 та ± 2216 виконати перетворення у внутрішній (машинний) формат подання чисел в пам'яті комп'ютера за умови, що числа є знаковими. Для збереження чисел використовується 8 та 16 бітів.

Використовуючи відомі правила переведу, спочатку переведемо задані числа в двійкову систему числення за допомогою додаткового коду, а потім дамо їх внутрішнє представлення. Результати зведемо в таблицю 2.3.

$$\begin{array}{r}
 21 \mid 2 \\
 \hline
 \begin{array}{r}
 20 \quad 10 \mid 2 \\
 \hline
 1 \quad 10 \quad 5 \mid 2 \\
 \hline
 \quad 0 \quad 4 \quad 2 \mid 2 \\
 \hline
 \quad \quad 1 \quad 2 \quad 1 \\
 \hline
 \quad \quad \quad 0
 \end{array}
 \end{array}$$

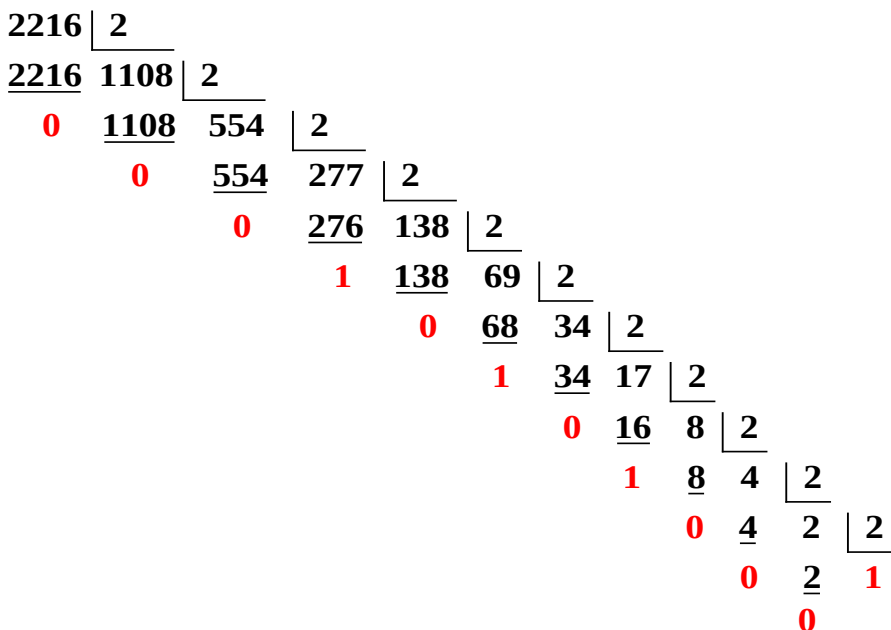
21d $\rightarrow$ 0001 0101b (BYTE)

- **21d $\rightarrow$ 1) 0001 0101b** - двійковий код числа $|-21|$

2) 1110 1010b - інверсія

3) $+$ 1

1110 1011b - додатковий код



2216d → 0000 1000 1010 1000b (WORD)

- 2216d → 1) 0000 1000 1010 1000b - двійковий код числа |-2216|

2) 1111 0111 0101 0111b - інверсія

3) + 1

1111 0111 0101 1000b - додатковий код

Таблиця 2.3

Машинне представлення заданих чисел

Dec	Byte		Word	
	Bin	Hex	Bin	Hex
21	0001.0101	15	0000.0000.0001.0101	0015
-21	1110.1011	EB	1111.1111.1110.1011	FFEB
2216			0000.1000.1010.1000	08A8
-2216			1111.0111.0101.1000	F758

Перевірка на assembler 5.0

Файл lr_02.asm

```
.MODEL tiny
.DATA
; lr_02
; ==== Byte ====
f0 db 21
; ==== Shortint ====
f1 db -21
; ==== Word ====
f2 dw 2216
; ==== Integer ====
f3 dw -2216
END
```

Файл лістингу lr_02.lst

Turbo Assembler		Version 5.0	01-21-22
22:17:56	Page 1		
lr_02.asm			
1	0000	.MODEL tiny	
2	0000	.DATA	
3			
4		; lr_02	
5			
6		; ==== Byte ====	
7	0000 15	f0 db 21	
8			
9		; ==== Shortint ====	
10	0001 EB	f1 db -21	
11			
12		; ==== Word ====	
13	0002 08A8	f2 dw 2216	
14			
15		; ==== Integer ====	
16	0004 F758	f3 dw -2216	
17			
18		END	

Приклад 2. Для заданого набору байтів 5A08 визначити які числові значення може містити цей набір, взятий у пам'яті комп'ютера.

Припустим що заданий набір байтів 5A08 – без-знаковий, то нам спочатку необхідно перевести його з шістнадцяткової системи числення у двійкову:

$$5A08_{16} = \underbrace{5}_{0101} \underbrace{A}_{1010} \underbrace{0}_{0000} \underbrace{8}_{1000} = 0101\ 1010\ 0000\ 1000_2,$$

а потім з шістнадцяткової у десяткову:

$$5A08_{16} = 5 \cdot 16^3 + 10 \cdot 16^2 + 8 \cdot 16^0 = 23048_{10},$$

або з двійкової у десяткову:

$$0101\ 1010\ 0000\ 1000_2 = 2^3 + 2^9 + 2^{11} + 2^{12} + 2^{14} = 23048_{10}$$

Але, якщо припустити що заданий набір байтів 5A08 – знаковий, то нам необхідно отримане двійкове значення інвертувати і додати одиницю до молодшого розряду для отримання додаткового коду:

1) 0101 1010 0000 1000 - двійковий код набору байтів 5A08

2) 1010 0101 1111 0111 - інверсія

3)
$$\begin{array}{r} + \\ \hline 1 \end{array}$$

1010 0101 1111 1000 - **додатковий код**

Отримане двійкове значення переводимо у десяткову систему числення:

$$1010\ 0101\ 1111\ 1000_2 = -42488_{10}$$

Виходячи з вище приведених розрахунків заданий набір байтів 5A08 може відповідати двом десятковим числам: 23048₁₀ та -42488₁₀.

Перевірка на assembler 5.0

```
Turbo Assembler          Version 5.0          01-21-22
22:27:30          Page 1
lr_02.asm
1  0000                      .MODEL tiny
2  0000                      .DATA
3
4                          ; lr_02
5
6                          ; ==== Integer ====
7  0000      5A08            f0      dw      23048
8  0002      5A08            dw      -42488
```

Приклад 3: Для заданого набору байтів D2C8 визначити які числові значення може містити додатний набір байтів згідно стандарту 1251 (ANSI, WIN).

Спочатку ділимо заданий набір байтів по два шістнадцяткових числа: D2 C8

Потім обираємо відповідні символи з таблиці 2.12. Причому, старші шістнадцяткові числа з кожного отриманного числа (D та C) будуть відповідати за рядки, молодші (2 та 8) – за стовпці:

D2 → Т

C8 → И

Заданий набір байтів D2C8 містить символи ТИ.

Завдання на лабораторну роботу

1. Для заданих чисел (табл.2.5) виконати перетворення у внутрішній (машинний) формат подання чисел в пам'яті комп'ютера за умови, що числа є:

- беззнаковими;
- знаковими.

Для збереження чисел використовується 8, 16 та 32 біти. Представити перевірки на мові assembler 5.0.

2. Для заданих чисел (табл.2.6) виконати перетворення у внутрішній(машинний) формат подання чисел в пам'яті комп'ютера за умови, що числа є:

- беззнаковими;
- знаковими.

Для збереження чисел використовується 16 та 32 біти. Представити перевірки на мові assembler 5.0.

3. Для заданих чисел (табл.2.7) виконати перетворення у внутрішній(машинний) формат подання чисел в пам'яті комп'ютера за умови, що числа є:

- беззнаковими;
- знаковими.

Для збереження чисел використовується 32 біти. Представити перевірки на мові assembler 5.0.

4. За даними табл. 2.8 визначити які числові значення може містити знаковий набір байтів, взятий у пам'яті комп'ютера. Представити перевірки на мові assembler 5.0.

5. За даними табл. 2.8 визначити які числові значення може містити додатний набір байтів згідно стандарту 1251 (ANSI, WIN)(табл.2.9).

Вимоги до оформлення звіту

Звіт оформлюється у текстовому документі і має містити:

1. Правильно оформлену рамку.
2. Тему, мету, номер варіанту.
3. Виконані **всі** завдання з ходом обрахунків і перевітками на мові Assmbler.
4. Висновки.

Звіт оформлений неналежним чином і без ходу обрахунків не оцінюється і має бути перероблений!

Дані для виконання завдання 1

№ варіанта	Число 1	Число 2
1	$\pm(30+G+N)$	$\pm(126-G-N)$
2	$\pm(19+G+N)$	$\pm(122-G-N)$
3	$\pm(23+G+N)$	$\pm(118-G-N)$
4	$\pm(14+G+N)$	$\pm(121-G-N)$
5	$\pm(6+G+N)$	$\pm(120-G-N)$
6	$\pm(16+G+N)$	$\pm(122-G-N)$
7	$\pm(28+G+N)$	$\pm(117-G-N)$
8	$\pm(8+G+N)$	$\pm(120-G-N)$
9	$\pm(21+G+N)$	$\pm(123-G-N)$
10	$\pm(10+G+N)$	$\pm(114-G-N)$
11	$\pm(24+G+N)$	$\pm(112-G-N)$
12	$\pm(4+G+N)$	$\pm(116-G-N)$
13	$\pm(27+G+N)$	$\pm(116-G-N)$
14	$\pm(9+G+N)$	$\pm(124-G-N)$
15	$\pm(20+G+N)$	$\pm(112-G-N)$
16	$\pm(1+G+N)$	$\pm(119-G-N)$
17	$\pm(29+G+N)$	$\pm(111-G-N)$
18	$\pm(12+G+N)$	$\pm(121-G-N)$
19	$\pm(26+G+N)$	$\pm(115-G-N)$
20	$\pm(3+G+N)$	$\pm(113-G-N)$
21	$\pm(17+G+N)$	$\pm(110-G-N)$
22	$\pm(25+G+N)$	$\pm(125-G-N)$
23	$\pm(11+G+N)$	$\pm(119-G-N)$
24	$\pm(15+G+N)$	$\pm(111-G-N)$
25	$\pm(2+G+N)$	$\pm(114-G-N)$
26	$\pm(22+G+N)$	$\pm(118-G-N)$
27	$\pm(7+G+N)$	$\pm(115-G-N)$
28	$\pm(18+G+N)$	$\pm(117-G-N)$
29	$\pm(5+G+N)$	$\pm(113-G-N)$
30	$\pm(13+G+N)$	$\pm(110-G-N)$

G – номер групи, N – номер варіанта

Дані для виконання завдання 2

№ варіанта	Число 1	Число 2
1	$\pm(10214+G+N)$	$\pm(10214-G-N)$
2	$\pm(11106+G+N)$	$\pm(11106-G-N)$
3	$\pm(10054+G+N)$	$\pm(10054-G-N)$
4	$\pm(11001+G+N)$	$\pm(11001-G-N)$
5	$\pm(10549+G+N)$	$\pm(10549-G-N)$
6	$\pm(10653+G+N)$	$\pm(10653-G-N)$
7	$\pm(10517+G+N)$	$\pm(10517-G-N)$
8	$\pm(11048+G+N)$	$\pm(11048-G-N)$
9	$\pm(10652+G+N)$	$\pm(10652-G-N)$
10	$\pm(10557+G+N)$	$\pm(10557-G-N)$
11	$\pm(11005+G+N)$	$\pm(11005-G-N)$
12	$\pm(10812+G+N)$	$\pm(10812-G-N)$
13	$\pm(11032+G+N)$	$\pm(11032-G-N)$
14	$\pm(10146+G+N)$	$\pm(10146-G-N)$
15	$\pm(11010+G+N)$	$\pm(11010-G-N)$
16	$\pm(10930+G+N)$	$\pm(10930-G-N)$
17	$\pm(10522+G+N)$	$\pm(10522-G-N)$
18	$\pm(10023+G+N)$	$\pm(10023-G-N)$
19	$\pm(10475+G+N)$	$\pm(10475-G-N)$
20	$\pm(10933+G+N)$	$\pm(10933-G-N)$
21	$\pm(10736+G+N)$	$\pm(10736-G-N)$
22	$\pm(11357+G+N)$	$\pm(11357-G-N)$
23	$\pm(10732+G+N)$	$\pm(10732-G-N)$
24	$\pm(10925+G+N)$	$\pm(10925-G-N)$
25	$\pm(11320+G+N)$	$\pm(11320-G-N)$
26	$\pm(10369+G+N)$	$\pm(10369-G-N)$
27	$\pm(10698+G+N)$	$\pm(10698-G-N)$
28	$\pm(10547+G+N)$	$\pm(10547-G-N)$
29	$\pm(10053+G+N)$	$\pm(10053-G-N)$
30	$\pm(10150+G+N)$	$\pm(10150-G-N)$

G – номер групи, N – номер варіанта

Дані для виконання завдання 3

№ варіанта	Число 1	Число 2
1	$\pm(32547+G+N)$	$\pm(32547-G-N)$
2	$\pm(32526+G+N)$	$\pm(32526-G-N)$
3	$\pm(32491+G+N)$	$\pm(32491-G-N)$
4	$\pm(32412+G+N)$	$\pm(32412-G-N)$
5	$\pm(32385+G+N)$	$\pm(32385-G-N)$
6	$\pm(32344+G+N)$	$\pm(32344-G-N)$
7	$\pm(32293+G+N)$	$\pm(32293-G-N)$
8	$\pm(32208+G+N)$	$\pm(32208-G-N)$
9	$\pm(32199+G+N)$	$\pm(32199-G-N)$
10	$\pm(32100+G+N)$	$\pm(32100-G-N)$
11	$\pm(31953+G+N)$	$\pm(31953-G-N)$
12	$\pm(31946+G+N)$	$\pm(31946-G-N)$
13	$\pm(31817+G+N)$	$\pm(31817-G-N)$
14	$\pm(31822+G+N)$	$\pm(31822-G-N)$
15	$\pm(31735+G+N)$	$\pm(31735-G-N)$
16	$\pm(31708+G+N)$	$\pm(31708-G-N)$
17	$\pm(31699+G+N)$	$\pm(31699-G-N)$
18	$\pm(31624+G+N)$	$\pm(31624-G-N)$
19	$\pm(31573+G+N)$	$\pm(31573-G-N)$
20	$\pm(31510+G+N)$	$\pm(31510-G-N)$
21	$\pm(30975+G+N)$	$\pm(30975-G-N)$
22	$\pm(30912+G+N)$	$\pm(30912-G-N)$
23	$\pm(30867+G+N)$	$\pm(30867-G-N)$
24	$\pm(30808+G+N)$	$\pm(30808-G-N)$
25	$\pm(30799+G+N)$	$\pm(30799-G-N)$
26	$\pm(30734+G+N)$	$\pm(30734-G-N)$
27	$\pm(30683+G+N)$	$\pm(30683-G-N)$
28	$\pm(30616+G+N)$	$\pm(30616-G-N)$
29	$\pm(30571+G+N)$	$\pm(30571-G-N)$
30	$\pm(30522+G+N)$	$\pm(30522-G-N)$

G – номер групи, N – номер варіанта

Дані для виконання завдання 4 та 5

№ варіанта	Число 1	Число 2
1	3F4A	EF489785
2	AB02	641B5A6A
3	130E	3962F148
4	2C36	62FC321D
5	4B78	987C6A56
6	E107	F698D645
7	7D95	E0330BD2
8	309B	34FE147E
9	85A0	5A5CC68A
10	C803	47DD8645
11	4A66	987D3A56
12	9B0E	951CCBD2
13	F032	6D096CE0
14	19C7	763C321D
15	7F41	6986D096
16	22AB	13E987D3
17	339E	CC68AB06
18	E12D	13E147EF
19	640F	F698F405
20	83EE	4053641B
21	D73C	68AB0631
22	A0CB	4E62F698
23	5AD1	E0330987
24	12AA	785A7895
25	C15F	45AE0330
26	EA07	34FEBD25
27	84DE	147EF364
28	A254	41BB0631
29	36BF	CE063641
30	1DC7	5A6A47DD

Таблиця 2.9

Таблиця символів 1251 (ANSI, WIN)

	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.A	.B	.C	.D	.E	.F
0.	0 ¹² 1	☺ 2	☻ 3	♥ 4	♦ 5	♣ 6	♠ 7	♣ 8	○ 9 ₃	◐ 10	♂ 11	♀ 12	♪ 13	♫ 14	✱ 15	
1.	▶ 16	◀ 17	↑ 18	!! 19	¶ 20	§ 21	— 22	↓ 23	↑ 24	↓ 25	→ 26	← 27	⌞ 28	↔ 29	▲ 30	▼ 31
2.	32	! 33	" 34	# 35	\$ 36	% 37	& 38	' 39	(40	) 41	* 42	+ 43	, 44	- 45	· 46	/ 47
3.	0 48	1 49	2 50	3 51	4 52	5 53	6 54	7 55	8 56	9 57	: 58	; 59	< 60	= 61	> 62	? 63
4.	@ 64	A 65	B 66	C 67	D 68	E 69	F 70	G 71	H 72	I 73	J 74	K 75	L 76	M 77	N 78	O 79
5.	P 80	Q 81	R 82	S 83	T 84	U 85	V 86	W 87	X 88	Y 89	Z 90	[91	\ 92	] 93	^ 94	_ 95
6.	` 96	a 97	b 98	c 99	d 100	e 101	f 102	g 103	h 104	i 105	j 106	k 107	l 108	m 109	n 110	o 111
7.	p 112	q 113	r 114	s 115	t 116	u 117	v 118	w 119	x 120	y 121	z 122	{ 123	 124	}	~ 126	△ 127
8.	Ђ 128	Ѓ 129	Ѕ 130	Ї 131	Љ 132	Њ 133	Ћ 134	Ќ 135	€ 136	‰ 137	Љ 138	‹ 139	Њ 140	Ќ 141	Ћ 142	Ќ 143
9.	ђ 144	‘ 145	’ 146	“ 147	” 148	• 149	— 150	— 151	152 ₄	™ 153	љ 154	› 155	њ 156	ќ 157	ћ 158	џ 159
A.	160 ₅	Ў 161	Ў 162	Ј 163	Ѱ 164	Ѓ 165	Ѕ 166	Ѕ 167	Ё 168	© 169	€ 170	« 171	¬ 172	173 ₆	® 174	Ѕ 175
B.	176	± 177	І 178	і 179	Ѓ 180	μ 181	¶ 182	· 183	ё 184	№ 185	€ 186	» 187	ј 188	Ѕ 189	Ѕ 190	і 191
C.	А 192	Б 193	В 194	Г 195	Д 196	Е 197	Ж 198	З 199	И 200	Й 201	К 202	Л 203	М 204	Н 205	О 206	П 207
D.	Р 208	С 209	Т 210	У 211	Ф 212	Х 213	Ц 214	Ч 215	Ш 216	Щ 217	Ъ 218	Ы 219	Ь 220	Э 221	Ю 222	Я 223

Продовження таблиці 2.12

Таблиця символів 1251 (ANSI, WIN)

Е.	а 224	б 225	в 226	г 227	д 228	е 229	ж 230	з 231	и 232	й 233	к 234	л 235	м 236	н 237	о 238	п 239
Ф.	р 240	с 241	т 242	у 243	ф 244	х 245	ц 246	ч 247	ш 248	щ 249	ъ 250	ы 251	ь 252	э 253	ю 254	я 255

Контрольні питання

1. Що таке тип даних?
2. Які існують типи даних?
3. Поняття цілого і дійсного числа. Приклади.
4. Поняття біт, байт, слово, подвійне слово, зчетверене слово.
5. Який розмір в байтах мають слово, подвійне слово, зчетверене слово?
6. Які директиви використовуються у мові програмування Асемблер?
7. Базові цілочислені типи даних для мови С та їх діапазони допустимих значень.
8. Правило формування знакових цілих чисел за допомогою додаткового коду.
9. Алгоритм внутрішнього(машинного) представлення цілочисельних беззнакових даних.
10. Алгоритм внутрішнього(машинного) представлення цілочисельних знакових даних.