Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки

**ЛАБОРАТОРНА РОБОТА №2**

з дисципліни «Системне програмування» на тему

«Внутрішнє представлення цілих і дійсних даних у процесорі ix86»

Виконав: Перевірив:

студент II курсу ФІОТ доц. Павлов В. Г. групи ІМ-22 Басараб Станіслав Анатолійович  
номер у списку групи: 2

Київ 2024

**Мета роботи**: Вивчення форматів внутрішнього представлення цілих і дійсних чисел шляхом порівняння результатів, отриманих теоретично, з результатами, що містяться в отриманому розширеному лістингу програми.

**Порядок виконання роботи**

1. Сформувати наступні початкові числа на основі дати народження студента, записаної у форматі **dd.mm.yyyy**:

* **A** – ціле двозначне число, що дорівнює **dd**;
* **B** – ціле чотиризначне число, яке дорівнює **ddmm**, записаним без роздільника;
* **C** – ціле восьмизначне число, яке дорівнює **ddmmyyyy**, записаним без роздільника;
* **D** – число, отримане шляхом ділення числа **A** на число **N**,  
  де **N** – чотиризначний номер залікової книжки;
* **E** – число, отримане шляхом ділення числа **B** на число **N**;
* **F** – число отримане шляхом ділення числа **C** на число **N**;

***(при діленні округлювати результат до 3 знаків після коми)***

1. Представити отримані числа в двійковій системі числення ***(дробову частину округлювати до 10 знаків після коми)***.
2. За допомогою розрядної сітки показати в звіті представлення цілих чисел в наступних форматах:

* **“ddmmyyyy”** у вигляді символьного рядка;
* числа **A** і **-A** у однобайтовому форматі **Byte**;
* числа **A**, **B**, **-A** і **-B** у двобайтовому форматі **Word**;
* числа **A**, **B**, **C**, **-A**, **-B** і **-C** у чотирьохбайтовому форматі **ShortInt**;
* числа **A**, **B**, **C**, **-A**, **-B** і **-C** у восьмибайтовому форматі **LongInt**;

1. Представити числа **D**, **E** і **F** у нормалізованому вигляді.
2. За допомогою розрядної сітки показати в звіті представлення дійсних чисел в наступних форматах:

* числа **D** і **-D** у чотирьохбайтовому форматі **Single** (float);
* числа **E** і **-E** у восьмибайтовому форматі **Double** (double);
* числа **F** і **-F** у десятибайтовому форматі **Extended** (long double).

1. Остаточні результати по кожному пункту надати у звіті окрім двійкової ще і в шістнадцятковій системі числення.
2. Скласти програму на мові **Assembler**, в якій задати вказані символи і усі варіанти констант **A, B, … F**, які присутні у пунктах 3 та 5. Значення констант **A, B, … F** вивести у віконній формі на екран.
3. У режимі консолі виконати компіляцію і виконання програми. Привести скріншот вікна виведення значень змінних в звіті.
4. У режимі консолі виконати формування розширеного лістингу програми у вигляді файлу з розширенням **“lst”** за допомогою опції **/FI** компілятора **ML**.
5. Знайти в лістингу фрагмент, що містить коди команд і даних, і привести цей фрагмент в звіті з виконання лаб. роботи.
6. У наведеному фрагменті лістингу знайти і помітити кольоровим олівцем або фломастером шістнадцяткові коди усіх варіантів представлення чисел **A, B, … F**.
7. Порівняти результати, що містяться в лістингу, з розрахунковими, отриманими при виконанні пунктів 3 – 6, і зробити висновки по лабораторній роботі.

**Виконання роботи**

Сформуємо початкові числа. Дата народження: **07.04.2005**

N (номер залікової книжки) = **8856**

1. A = 07
2. B = 0704
3. C = 07042005
4. D = A/N = 07/8856 ≈ 0,001
5. E = B/N = 0704/8856 ≈ 0,079
6. F = C/N = 07042005/8856 ≈ 795,168

Представимо отримані числа в двійковій системі числення:

1. A = 0710 = 1112
2. B = 070410 = 10110000002
3. C = 0704200510 = 110101101110011110101012
4. D = 0,00110:

010 = 02

|  |  |  |
| --- | --- | --- |
| ,001 \* 2 = | 0 | ,002 |
| ,002 \* 2 = | 0 | ,004 |
| ,004 \* 2 = | 0 | ,008 |
| ,008 \* 2 = | 0 | ,016 |
| ,016 \* 2 = | 0 | ,032 |
| ,032 \* 2 = | 0 | ,064 |
| ,064 \* 2 = | 0 | ,128 |
| ,128 \* 2 = | 0 | ,256 |
| ,256 \* 2 = | 0 | ,512 |
| ,512 \* 2 = | 1 | ,024 |
| ,024 \* 2 = | 0 | ,048 |

,00110 = ,00000000012

0,00110 = 0,00000000012

1. E = 0,07910:

010 = 02

|  |  |  |
| --- | --- | --- |
| ,079 \* 2 = | 0 | ,158 |
| ,158 \* 2 = | 0 | ,316 |
| ,316 \* 2 = | 0 | ,632 |
| ,632 \* 2 = | 1 | ,264 |
| ,264 \* 2 = | 0 | ,528 |
| ,528 \* 2 = | 1 | ,056 |
| ,056 \* 2 = | 0 | ,112 |
| ,112 \* 2 = | 0 | ,224 |
| ,224 \* 2 = | 0 | ,448 |
| ,448 \* 2 = | 0 | ,896 |
| ,896 \* 2 = | 1 | ,792 |

Оскільки одинадцятий знак після коми = 1, проводимо округлення

,07910 = ,00010100012

0,07910 = 0,00010100012

1. F = 795,16810:

79510 = 11000110112

|  |  |  |
| --- | --- | --- |
| ,168 \* 2 = | 0 | ,336 |
| ,336 \* 2 = | 0 | ,672 |
| ,672 \* 2 = | 1 | ,344 |
| ,344 \* 2 = | 0 | ,688 |
| ,688 \* 2 = | 1 | ,376 |
| ,376 \* 2 = | 0 | ,752 |
| ,752 \* 2 = | 1 | ,504 |
| ,504 \* 2 = | 1 | ,008 |
| ,008 \* 2 = | 0 | ,016 |
| ,016 \* 2 = | 0 | ,032 |
| ,032 \* 2 = | 0 | ,064 |

,16810 = ,00101011002

795,16810 = 1100011011,00101011002

За допомогою розрядної сітки представляємо цілі числа в наступних форматах:

1. “ddmmyyyy” у вигляді символьного рядка (ASCII)

“0” = 3016

“7” = 3716

“0” = 3016

“4” = 3416

“2” = 3216

“0” = 3016

“0” = 3016

“5” = 3516

1. числа A і -A у однобайтовому форматі Byte

* A:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

A = 0710 = 000001112 = 0716

* -A:

Інвертуємо А

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |

Додаємо 1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |

-A = -0710 = 111110012 = F916

1. числа A, B, -A і -B у двобайтовому форматі Word

* A:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

A = 0710 = 00000000000001112 = 000716

* B:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

B = 070410 = 00000010110000002 = 02C016

* -A:

Інвертуємо A

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |

Додаємо 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |

-A = -0710 = 11111111111110012 = FFF916

* -B:

Інвертуємо B

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |

Додаємо 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

-B = -070410 = 11111101010000002 = FD4016

1. числа A, B, C, -A, -B і -C у чотирьохбайтовому форматі ShortInt

* A:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

A = 0710 = 0000 0000 0000 0000 0000 0000 0000 01112 = 00 00 00 0716

* B:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

B = 070410 = 0000 0000 0000 0000 0000 0010 1100 00002 = 00 00 02 C016

* C:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

С = 0704200510 = 0000 0000 0110 1011 0111 0011 1101 01012 =   
= 00 6B 73 D516

* -A:

Інвертуємо A

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |

Додаємо 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |

-A = -0710 = 1111 1111 1111 1111 1111 1111 1111 10012 = FF FF FF F916

* -B:

Інвертуємо B

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |

Додаємо 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

-B = -070410 = 1111 1111 1111 1111 1111 1101 0100 00002 = FF FF FD 4016

* -C:

Інвертуємо С

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |

Додаємо 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |

-С = -0704200510 = 1111 1111 1001 0100 1000 1100 0010 10112 =  
= FF 94 8C 2B16

1. числа A, B, C, -A, -B і -C у восьмибайтовому форматі LongInt

* A:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0111 |

A = 0710 = 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 01112 = 00 00 00 00 00 00 00 0716

* B:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0010 | 1100 | 0000 |

B = 070410 = 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0010 1100 00002 = 00 00 00 00 00 00 02 C016

* C:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0110 | 1011 | 0111 | 0011 | 1101 | 0101 |

С = 0704200510 = 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0110 1011 0111 0011 1101 01012 = 00 00 00 00 00 6B 73 D516

* -A:

Інвертуємо A

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1000 |

Додаємо 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1001 |

-A = -0710 = 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 10012 = FF FF FF FF FF FF FF F916

* -B:

Інвертуємо B

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1101 | 0011 | 1111 |

Додаємо 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1101 | 0100 | 0000 |

-B = -070410 = 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1101 0100 00002 = FF FF FF FF FF FF FD 4016

* -C:

Інвертуємо C

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1001 | 0100 | 1000 | 1100 | 0010 | 1010 |

Додаємо 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1001 | 0100 | 1000 | 1100 | 0010 | 1011 |

-С = -0704200510 = 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1001 0100 1000 1100 0010 10112 = FF FF FF FF FF 94 8C 2B16

Представимо числа D, E і F у нормалізованому вигляді:

1. D:

* D = 0,00110 = 0,00000000012
* 0,00000000012 = 1 \* 2-10
* ZN=+0, M = 1, N = 2, q = -10

1. E:

* E = 0,07910 = 0,00010100012
* 0,00010100012 = 1,010001 \* 2-4
* ZN=+0, M = 1,010001, N = 2, q = -4

1. F:

* F = 795,16810 = 1100011011,00101011002
* 1100011011,00101011002 = 1,1000110110010101100 \* 29
* ZN=+0, M = 1,1000110110010101100, N = 2, q = 9

За допомогою розрядної сітки представляємо дійсні числа в наступних форматах:

1. числа D і -D у чотирьохбайтовому форматі Single (float):

* D = 0,00110

1. Знак числа – 0 (додатній)
2. Порядок: q + 127 = -10 + 127 = 11710 = 11101012
3. Мантиса: 1 → 0
4. Розрядна сітка:

|  |  |  |
| --- | --- | --- |
| 31 | 30 23 | 22 0 |
| **0** | **01110101** | **00000000000000000000000** |
| Знак | Порядок | Мантиса |

1. Отже: 0011 1010 1000 0000 0000 0000 0000 0000 =  
   = 3A 80 00 0016

* -D = -0,00110

1. Знак числа – 1 (від’ємний)
2. Порядок: q + 127 = -10 + 127 = 11710 = 11101012
3. Мантиса: 1 → 0
4. Розрядна сітка:

|  |  |  |
| --- | --- | --- |
| 31 | 30 23 | 22 0 |
| **1** | **01110101** | **00000000000000000000000** |
| Знак | Порядок | Мантиса |

1. Отже: 1011 1010 1000 0000 0000 0000 0000 0000 =  
   = BA 80 00 0016
2. числа E і -E у восьмибайтовому форматі Double (double):

* E = 0,07910

1. Знак числа – 0 (додатній)
2. Порядок: q + 1023 = -4 + 1023 = 101910 = 11111110112
3. Мантиса: 1,010001 → 010001
4. Розрядна сітка:

|  |  |  |
| --- | --- | --- |
| 63 | 62 52 | 51 0 |
| **0** | **01111111011** | **0100010000000000000000000000000000000000000000000000** |
| Знак | Порядок | Мантиса |

1. Отже: 0011 1111 1011 0100 0100 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 = 3F B4 40 00 00 00 00 0016

* -E = -0,07910

1. Знак числа – 1 (від’ємний)
2. Порядок: q + 1023 = -4 + 1023 = 101910 = 11111110112
3. Мантиса: 1,010001 → 010001
4. Розрядна сітка:

|  |  |  |
| --- | --- | --- |
| 63 | 62 52 | 51 0 |
| **1** | **01111111011** | **0100010000000000000000000000000000000000000000000000** |
| Знак | Порядок | Мантиса |

1. Отже: 1011 1111 1011 0100 0100 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 = BF B4 40 00 00 00 00 0016
2. числа F і -F у десятибайтовому форматі Extended (long double):

* F = 795,16810

1. Знак числа – 0 (додатній)
2. Порядок: q + 16383 = 9 + 16383 = 1639210 = 1000000000010002
3. Мантиса: 1,1000110110010101100 → 1000110110010101100
4. Розрядна сітка:

|  |  |  |
| --- | --- | --- |
| 79 | 78 64 | 63 0 |
| **0** | **100000000001000** | **1000110110010101100000000000000000000000000000000000000000000000** |
| Знак | Порядок | Мантиса |

1. Отже: 0100 0000 0000 1000 1000 1101 1001 0101 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 = 40 08 8D 95 80 00 00 00 00 0016

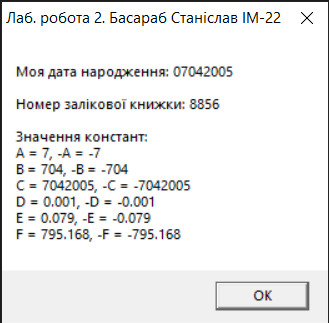
* -F = -795,16810

1. Знак числа – 1 (від’ємний)
2. Порядок: q + 16383 = 9 + 16383 = 1639210 = 1000000000010002
3. Мантиса: 1,1000110110010101100 → 1000110110010101100
4. Розрядна сітка:

|  |  |  |
| --- | --- | --- |
| 79 | 78 64 | 63 0 |
| **1** | **100000000001000** | **1000110110010101100000000000000000000000000000000000000000000000** |
| Знак | Порядок | Мантиса |

Отже: 1100 0000 0000 1000 1000 1101 1001 0101 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 = C0 08 8D 95 80 00 00 00 00 0016

Скріншот вікна виведення значень змінних



Фрагмент лістингу програми з виділеними 16-ковими кодами чисел A, B, … F

00000000 .data?

00000000 00000100 [ windowContentBuffer db 256 dup (?)

00

]

00000100 00000020 [ positiveDStringBuffer db 32 dup (?)

00

]

00000120 00000020 [ negativeDStringBuffer db 32 dup (?)

00

]

00000140 00000020 [ positiveEStringBuffer db 32 dup (?)

00

]

00000160 00000080 [ negativeEStringBuffer db 128 dup (?)

00

]

000001E0 00000080 [ positiveFStringBuffer db 128 dup (?)

00

]

00000260 00000080 [ negativeFStringBuffer db 128 dup (?)

00

]

00000000 .data

00000000 D0 9B D0 B0 D0 windowCaption db "╨Ы╨░╨▒. ╤А╨╛╨▒╨╛╤В╨░ 2. ╨С╨░╤Б╨░╤А╨░╨▒ ╨б╤В╨░╨╜╤Ц╤Б╨╗╨░╨▓ ╨Ж╨Ь-22", 0

B1 2E 20 D1

80 D0 BE D0

B1 D0 BE D1

82 D0 B0 20

32 2E 20 D0

91 D0 B0 D1

81 D0 B0 D1

80 D0 B0 D0

B1 20 D0 A1

D1 82 D0 B0

D0 BD D1 96

D1 81 D0 BB

D0 B0 D0 B2

20 D0 86 D0

9C 2D 32 32

00

00000042 D0 9C D0 BE D1 windowContent db "╨Ь╨╛╤П ╨┤╨░╤В╨░ ╨╜╨░╤А╨╛╨┤╨╢╨╡╨╜╨╜╤П: 07042005", 10, 10,

8F 20 D0 B4

D0 B0 D1 82

D0 B0 20 D0

BD D0 B0 D1

80 D0 BE D0

B4 D0 B6 D0

B5 D0 BD D0

BD D1 8F 3A

20 30 37 30

34 32 30 30

35 0A 0A D0

9D D0 BE D0

BC D0 B5 D1

80 20 D0 B7

D0 B0 D0 BB

D1 96 D0 BA

D0 BE D0 B2

D0 BE D1 97

20 D0 BA D0

BD D0 B8 D0

B6 D0 BA D0

B8 3A 20 38

38 35 36 0A

0A D0 97 D0

BD D0 B0 D1

87 D0 B5 D0

BD D0 BD D1

8F 20 D0 BA

D0 BE D0 BD

D1 81 D1 82

D0 B0 D0 BD

D1 82 3A 0A

41 20 3D 20

25 64 2C 20

2D 41 20 3D

20 25 64 0A

42 20 3D 20

25 64 2C 20

2D 42 20 3D

20 25 64 0A

43 20 3D 20

25 64 2C 20

2D 43 20 3D

20 25 64 0A

44 20 3D 20

25 73 2C 20

2D 44 20 3D

20 25 73 0A

45 20 3D 20

25 73 2C 20

2D 45 20 3D

20 25 73 0A

46 20 3D 20

25 73 2C 20

2D 46 20 3D

20 25 73 00

"╨Э╨╛╨╝╨╡╤А ╨╖╨░╨╗╤Ц╨║╨╛╨▓╨╛╤Ч ╨║╨╜╨╕╨╢╨║╨╕: 8856", 10, 10,

"╨Ч╨╜╨░╤З╨╡╨╜╨╜╤П ╨║╨╛╨╜╤Б╤В╨░╨╜╤В:", 10,

"A = %d, -A = %d", 10,

"B = %d, -B = %d", 10,

"C = %d, -C = %d", 10,

"D = %s, -D = %s", 10,

"E = %s, -E = %s", 10,

"F = %s, -F = %s", 0

00000127 30 37 30 34 32 dateOfBirth db "07042005", 0

30 30 35 00

00000130 07 positiveAByte db 07

00000131 F9 negativeAByte db -07

00000132 0007 positiveAWord dw 07

00000134 02C0 positiveBWord dw 0704

00000136 FFF9 negativeAWord dw -07

00000138 FD40 negativeBWord dw -0704

0000013A 00000007 positiveAShortInt dd 07

0000013E 000002C0 positiveBShortInt dd 0704

00000142 006B73D5 positiveCShortInt dd 07042005

00000146 FFFFFFF9 negativeAShortInt dd -07

0000014A FFFFFD40 negativeBShortInt dd -0704

0000014E FF948C2B negativeCShortInt dd -07042005

00000152 3A83126F positiveDSingle dd 0.001

00000156 BA83126F negativeDSingle dd -0.001

0000015A positiveALongInt dq 07

0000000000000007

00000162 positiveBLongInt dq 0704

00000000000002C0

0000016A positiveCLongInt dq 07042005

00000000006B73D5

00000172 negativeALongInt dq -07

FFFFFFFFFFFFFFF9

0000017A negativeBLongInt dq -0704

FFFFFFFFFFFFFD40

00000182 negativeCLongInt dq -07042005

FFFFFFFFFF948C2B

0000018A positiveDDouble dq 0.001

3F50624DD2F1A9FC

00000192 positiveEDouble dq 0.079

3FB4395810624DD3

0000019A positiveFDouble dq 795.168

4088D95810624DD3

000001A2 negativeDDouble dq -0.001

BF50624DD2F1A9FC

000001AA negativeEDouble dq -0.079

BFB4395810624DD3

000001B2 negativeFDouble dq -795.168

C088D95810624DD3

000001BA positiveFExtended dt 795.168

4008C6CAC083126E978D

000001C4 negativeFExtended dt -795.168

C008C6CAC083126E978D

00000000 .code

00000000 start:

invoke FloatToStr2, positiveDDouble, addr positiveDStringBuffer

invoke FloatToStr2, positiveEDouble, addr positiveEStringBuffer

invoke FloatToStr2, positiveFDouble, addr positiveFStringBuffer

invoke FloatToStr2, negativeDDouble, addr negativeDStringBuffer

invoke FloatToStr2, negativeEDouble, addr negativeEStringBuffer

invoke FloatToStr2, negativeFDouble, addr negativeFStringBuffer

invoke wsprintf,

addr windowContentBuffer,

addr windowContent,

positiveAShortInt,

negativeAShortInt,

positiveBShortInt,

negativeBShortInt,

positiveCShortInt,

negativeCShortInt,

addr positiveDStringBuffer,

addr negativeDStringBuffer,

addr positiveEStringBuffer,

addr negativeEStringBuffer,

addr positiveFStringBuffer,

addr negativeFStringBuffer

invoke MessageBox, NULL, addr windowContentBuffer, addr windowCaption, MB\_OK

invoke ExitProcess, 0

end start

Порівняння даних лістингу з розрахунковими

|  |  |  |  |
| --- | --- | --- | --- |
| Константа (формат) | Дані з лістингу | Результат розрахунків | Порівняння |
| “ddmmyyyy” | 30 37 30 34 32 30 30 35 | “0” = 3016  “7” = 3716  “0” = 3016  “4” = 3416  “2” = 3216  “0” = 3016  “0” = 3016  “5” = 3516 | Результат розрахунків співпадає з даними з лістингу |
| A (Byte) | 07 | 0716 | Результат розрахунків співпадає з даними з лістингу |
| -A (Byte) | F9 | F916 | Результат розрахунків співпадає з даними з лістингу |
| A (Word) | 0007 | 000716 | Результат розрахунків співпадає з даними з лістингу |
| B (Word) | 02C0 | 02C016 | Результат розрахунків співпадає з даними з лістингу |
| -A (Word) | FFF9 | FFF916 | Результат розрахунків співпадає з даними з лістингу |
| -B (Word) | FD40 | FD4016 | Результат розрахунків співпадає з даними з лістингу |
| A (ShortInt) | 00000007 | 00 00 00 0716 | Результат розрахунків співпадає з даними з лістингу |
| B (ShortInt) | 000002C0 | 00 00 02 C016 | Результат розрахунків співпадає з даними з лістингу |
| C (ShortInt) | 006B73D5 | 00 6B 73 D516 | Результат розрахунків співпадає з даними з лістингу |
| -A (ShortInt) | FFFFFFF9 | FF FF FF F916 | Результат розрахунків співпадає з даними з лістингу |
| -B (ShortInt) | FFFFFD40 | FF FF FD 4016 | Результат розрахунків співпадає з даними з лістингу |

|  |  |  |  |
| --- | --- | --- | --- |
| -C (ShortInt) | FF948C2B | FF 94 8C 2B16 | Результат розрахунків співпадає з даними з лістингу |
| A (LongInt) | 0000000000000007 | 00 00 00 00 00 00 00 0716 | Результат розрахунків співпадає з даними з лістингу |
| B (LongInt) | 00000000000002C0 | 00 00 00 00 00 00 02 C016 | Результат розрахунків співпадає з даними з лістингу |
| C (LongInt) | 00000000006B73D5 | 00 00 00 00 00 6B 73 D516 | Результат розрахунків співпадає з даними з лістингу |
| -A (LongInt) | FFFFFFFFFFFFFFF9 | FF FF FF FF FF FF FF F916 | Результат розрахунків співпадає з даними з лістингу |
| -B (LongInt) | FFFFFFFFFFFFFD40 | FF FF FF FF FF FF FD 4016 | Результат розрахунків співпадає з даними з лістингу |
| -C (LongInt) | FFFFFFFFFF948C2B | FF FF FF FF FF 94 8C 2B16 | Результат розрахунків співпадає з даними з лістингу |
| D (Single) | 3A83126F | **3A 8**0 00 0016 | Співпадають тільки перші три символи, тому що ми округлювали число до 10 знаків після коми |
| -D (Single) | BA83126F | **BA 8**0 00 0016 | Співпадають тільки перші три символи, тому що ми округлювали число до 10 знаків після коми |

|  |  |  |  |
| --- | --- | --- | --- |
| E (Double) | 3FB4395810624DD3 | **3F B4** 40 00 00 00 00 0016 | Співпадають тільки перші чотири символи, тому що ми округлювали число до 10 знаків після коми |
| -E (Double) | BFB4395810624DD3 | **BF B4** 40 00 00 00 00 0016 | Співпадають тільки перші чотири символи, тому що ми округлювали число до 10 знаків після коми |
| F (Extended) | 4008C6CAC083126E978D | **40 08** 8D 95 80 00 00 00 00 0016 | Співпадають тільки перші чотири символи, тому що ми округлювали число до 10 знаків після коми |
| -F (Extended) | C008C6CAC083126E978D | **C0 08** 8D 95 80 00 00 00 00 0016 | Співпадають тільки перші чотири символи, тому що ми округлювали число до 10 знаків після коми |

**Висновок**

Отже, у ході виконання лабораторної роботи була досягнута поставлена мета, а саме – вивчення форматів внутрішнього представлення цілих і дійсних чисел. У лістингу програми були знайдені на виділені шістнадцяткові коди представлення чисел.

Детальний аналіз розширеного лістингу програми дозволив порівняти отримані дані з теоретичними розрахунками. За результатами порівняння можна побачити, що представлення символьного рядку (ASCII) в лістингу повністю ідентичне до розрахунків. Так само можна й сказати про цілі числа, які в точності відповідають розрахункам. Проте варто зазначити, що дробові числа співпадають лише на декілька символів. Це говорить про те, що точність їх розрахунків залежить від того, на скільки знаків після коми ми округлюємо число при переведенні в іншу систему числення.

Таким чином, лабораторна робота допомогла у розширенні знань з системного програмування та виробленні практичних навичок у роботі з внутрішнім представленням чисел в програмному коді мови Асемблер.