**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 №1

з дисципліни «Системне програмування» на тему

“Внутрішнє представлення цілих

і дійсних даних у процесорі ix86”

Виконав: Перевірив:

Студент ІІ курсу ФІОТ доц. Павлов В. Г.

групи ІМ-12

Сутулов Нікіта Олегович

номер у списку групи (варіант): 20

Київ 2023

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

*Номер моєї залікової книжки – 1229, дата мого народження – 12.06.2004.*

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

* A – ціле двозначне число, що дорівнює dd;

A = 12

* B – ціле чотиризначне число, яке дорівнює ddmm, записаним без роздільника;

B = 1206

* C – ціле восьмизначне число, яке дорівнює ddmmyyyy, записаним без роздільника;

C = 12062004

Подальші числа D, E, F отримано шляхом ділення чисел A, B і C відповідно на номер залікової книжки, тобто є дійсними числами, які за завданням мають бути округлені до 3 знаків після коми:

* D – число, отримане шляхом ділення числа A на число N, де N тут і надалі - чотиризначний номер залікової книжки;

D = 12 / 1229 = 0.00976403580146460537021969080553 ≈ 0,010

* E – число, отримане шляхом ділення числа B на число N;

E = 1206 / 1229 = 0.98128559804719283970707892595606 ≈ 0,981

* F - число, отримане шляхом ділення числа C на число N;

F = 12062004 / 1229 = 9,814.4865744507729861676159479251 ≈

≈ 9814,487

**2. Представити отримані числа в двійковій системі числення (дробову частину округлювати до 10 знаків після коми).**

* A = 1210 = 11002
* B = 120610 = 100101101102
* C = 1206200410 = 1011100000001101001101002
* D = 0,01010:

010 = 02

| ,010 \* 2 = | 0 | ,020 |
| --- | --- | --- |
| ,020 \* 2 = | 0 | ,040 |
| ,040 \* 2 = | 0 | ,080 |
| ,080 \* 2 = | 0 | ,160 |
| ,160 \* 2 = | 0 | ,320 |
| ,320 \* 2 = | 0 | ,640 |
| ,640 \* 2 = | 1 | ,280 |
| ,280 \* 2 = | 0 | ,560 |
| ,560 \* 2 = | 1 | ,120 |
| ,120 \* 2 = | 0 | ,240 |
| ,240 \* 2 = | 0 | ,480 |

Так як одинадцятий знак після коми дорівнює 0, інші 10 знаків залишаємо, як є.

,01010 = ,00000010102

0,01010 = 0,00000010102

* E = 0,98110:

010 = 02

| ,981 \* 2 = | 1 | ,962 |
| --- | --- | --- |
| ,962 \* 2 = | 1 | ,924 |
| ,924 \* 2 = | 1 | ,848 |
| ,848 \* 2 = | 1 | ,696 |
| ,696 \* 2 = | 1 | ,392 |
| ,392 \* 2 = | 0 | ,784 |
| ,784 \* 2 = | 1 | ,568 |
| ,568 \* 2 = | 1 | ,136 |
| ,136 \* 2 = | 0 | ,272 |
| ,272 \* 2 = | 0 | ,544 |
| ,544 \* 2 = | 1 | ,088 |

Так як одинадцятий знак після коми дорівнює 1, округлюємо.

,98110 = ,11111011012

0,98110 = 0,11111011012

* F = 9814,48710:

981410 = 100110010101102

| ,487 \* 2 = | 0 | ,974 |
| --- | --- | --- |
| ,974 \* 2 = | 1 | ,948 |
| ,948 \* 2 = | 1 | ,896 |
| ,896 \* 2 = | 1 | ,792 |
| ,792 \* 2 = | 1 | ,584 |
| ,584 \* 2 = | 1 | ,168 |
| ,168 \* 2 = | 0 | ,336 |
| ,336 \* 2 = | 0 | ,672 |
| ,672 \* 2 = | 1 | ,344 |
| ,344 \* 2 = | 0 | ,688 |
| ,688 \* 2 = | 1 | ,376 |

Так як одинадцятий знак після коми дорівнює 1, округлюємо.

,48710 = ,01111100112

9814,48710 = 10011001010110,01111100112

**3. За допомогою розрядної сітки показати у звіті представлення цілих чисел в наступних форматах:**

* “ddmmyyyy” у вигляді символьного рядка:

C = 12062004

“1” = 3116

“2” = 3216

“0” = 3016

“6” = 3616

“2” = 3216

“0” = 3016

“0” = 3016

“4” = 3416

* числа A і –A в однобайтовому форматі Byte:
  + A:

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

A = 1210 = 000011002 = 0C16

* + -A:

1. інвертуємо A:

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

1. додаємо до інвертованого A одиницю:

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

-A = -1210 = 111101002 = F416

* числа A, B, -A і -B у двобайтовому форматі Word:
  + A:

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

A = 1210 = 00000000000011002 = 000C16

* + B:

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

B = 120610 = 00000100101101102 = 04B616

* + -A:

1. інвертуємо A:

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

1. додаємо до інвертованого A одиницю:

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

-A = -1210 = 11111111111101002 = FFF416

* + -B:

1. інвертуємо B:

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

1. додаємо до інвертованого B одиницю:

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

-B = -120610 = 11111011010010102 = FB4A16

* числа A, B, C, -A, -B і -C у чотирибайтовому форматі Shortlnt:
  + 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 | 1 | 1 | 0 | 0 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

A = 1210 = 0000 0000 0000 0000 0000 0000 0000 11002 = 00 00 00 0C16

* + B:

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

B = 120610 = 0000 0000 0000 0000 0000 0100 1011 01102 = 00 00 04 B616

* + C:

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

C = 1206200410 = 0000 0000 1011 1000 0000 1101 0011 01002 = 00 B8 0D 3416

* + -A:

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

-A = -1210 = 1111 1111 1111 1111 1111 1111 1111 01002 = FF FF FF F416

* + -B:

1. інвертуємо B:

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

-B = -120610 = 1111 1111 1111 1111 1111 1011 0100 10102 = FF FF FB 4A16

* + -C:

1. інвертуємо C:

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

1. додаємо 1:

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

-C = -1206200410 = 1111 1111 0100 0111 1111 0010 1100 11002 =

= FF 47 F2 CC16

* числа A, B, C, -A, -B і -C у восьмибайтовому форматі Longlnt:
  + A:

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

A = 1210 = 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 11002 = 00 00 00 00 00 00 00 0C16

* + B:

| 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0100 | 1011 | 0110 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

B = 120610 = 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0100 1011 01102 = 00 00 00 00 00 00 04 B616

* + C:

| 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 1011 | 1000 | 0000 | 1101 | 0011 | 0100 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

C = 1206200410 = 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 1011 1000 0000 1101 0011 01002 = 00 00 00 00 00 B8 0D 3416

* + -A:

1. інвертуємо A:

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

1. додаємо 1:

| 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 0100 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

-A = -1210 = 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 01002 = FF FF FF FF FF FF FF F416

* + -B:

1. інвертуємо B:

| 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1011 | 0100 | 1001 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

1. додаємо 1:

| 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1011 | 0100 | 1010 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

-B = -120610 = 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1011 0100 10102 = FF FF FF FF FF FF FB 4A16

* + -C:

1. інвертуємо C:

| 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 0100 | 0111 | 1111 | 0010 | 1100 | 1011 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

1. додаємо 1:

| 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 0100 | 0111 | 1111 | 0010 | 1100 | 1100 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

-C = -1206200410 = 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 0100 0111 1111 0010 1100 11002 = FF FF FF FF FF 47 F2 CC16

**4. Представити числа D, E і F у нормалізованому вигляді.**

* D

1. D = 0,01010 = 0,00000010102
2. 0,00000010102 = 1,010 \* 2-7
3. ZN=+0, M = 1,010, N = 2, q = -7

* E

1. E = 0,98110 = 0,11111011012
2. 0,11111011012 = 1,111101101 \* 2-1
3. ZN=+0, M = 1,111101101, N = 2, q = -1

* F

1. F = 9814,48710 = 10011001010110,01111100112
2. 10011001010110,01111100112 = 1,00110010101100111110011 \* 213
3. ZN=+0, M = 1,00110010101100111110011, N = 2, q = 13

**5. За допомогою розрядної сітки показати у звіті представлення дійсних чисел в наступних форматах:**

* числа D і -D в у чотирибайтовому форматі Single (float):
  + D = 0,01010

1. знак числа – 0 (додатній)
2. порядок: q + 127 = -7 + 127 = 12010 = 11110002
3. мантиса: 1,010 → 010
4. розрядна сітка:

| 31 | 30 23 | 22 0 |
| --- | --- | --- |
| 0 | 01111000 | 01000000000000000000000000 |
| Знак | Порядок | Мантиса |

1. отже:

0011 1100 0010 0000 0000 0000 0000 00002 = 3C 20 00 0016

* + -D = -0,01010

1. знак числа – 1 (від’ємний)
2. порядок: q + 127 = -7 + 127 = 12010 = 11110002
3. мантиса: 1,010 → 010
4. розрядна сітка:

| 31 | 30 23 | 22 0 |
| --- | --- | --- |
| 1 | 01111000 | 01000000000000000000000000 |
| Знак | Порядок | Мантиса |

1. отже:

1011 1100 0010 0000 0000 0000 0000 00002 = BC 20 00 0016

* числа E і -E у восьмибайтовому форматі Double (double):
  + E = 0,98110

1. знак числа – 0 (додатній)
2. порядок: q + 1023 = -1 + 1023 = 102210 = 11111111102
3. мантиса: 1,111101101 → 111101101
4. розрядна сітка:

| 63 | 62 52 | 51 0 |
| --- | --- | --- |
| 0 | 01111111110 | 1111011010000000000000000000000000000000000000000000 |
| Знак | Порядок | Мантиса |

1. отже:

0011 1111 1110 1111 0110 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 00002 =

= 3F EF 68 00 00 00 00 0016

* + -E = -0,98110

1. знак числа – 1 (від’ємний)
2. порядок: q + 1023 = -1 + 1023 = 102210 = 11111111102
3. мантиса: 1,111101101 → 111101101
4. розрядна сітка:

| 63 | 62 52 | 51 0 |
| --- | --- | --- |
| 0 | 01111111110 | 1111011010000000000000000000000000000000000000000000 |
| Знак | Порядок | Мантиса |

1. отже:

1011 1111 1110 1111 0110 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 00002 =

= BF EF 68 00 00 00 00 0016

* числа F и –F у десятибайтовому форматі Extended (long double):
  + F = 9814,48710

1. знак числа – 0 (додатній)
2. порядок: q + 16383 = 13 + 16383 = 1639610 = 1000000000011002
3. мантиса: 1,00110010101100111110011 → 100110010101100111110011
4. розрядна сітка:

| 79 | 78 64 | 63 0 |
| --- | --- | --- |
| 0 | 100000000001100 | 1001100101011001111100110000000000000000000000000000000000000000 |
| Знак | Порядок | Мантиса |

1. отже:

010000000000110010011001010110011111001100000000000000000000000000000000000000002 =

= 40 0C 99 59 F3 00 00 00 00 0016

* + -F = -9814,48710

1. знак числа – 1 (від’ємний)
2. порядок: q + 16383 = 13 + 16383 = 1639610 = 1000000000011002
3. мантиса: 1,00110010101100111110011 → 100110010101100111110011
4. розрядна сітка:

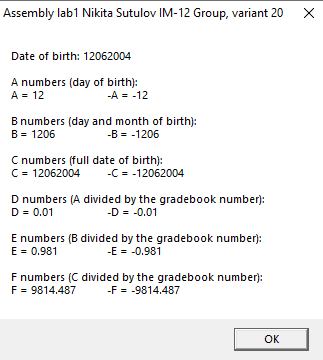
| 79 | 78 64 | 63 0 |
| --- | --- | --- |
| 1 | 100000000001100 | 1001100101011001111100110000000000000000000000000000000000000000 |
| Знак | Порядок | Мантиса |

1. отже:

110000000000110010011001010110011111001100000000000000000000000000000000000000002 =

= C0 0C 99 59 F3 00 00 00 00 0016

**6. Показати у звіті скриншот виконання програми:**



**7. Показати файл лістингу у звіті та виділити 16-кові коди чисел A, B, … F:**

00000000 .data

; Date of birth character string

00000000 31 32 30 36 32 SutulovDateOfBirthCharacterString db "12062004", 0

30 30 34 00

; Byte format

00000009 0C SutulovAPositiveByte db 12

0000000A F4 SutulovANegativeByte db -12

; Word format

0000000B 000C SutulovAPositiveWord dw 12

0000000D FFF4 SutulovANegativeWord dw -12

0000000F 04B6 SutulovBPositiveWord dw 1206

00000011 FB4A SutulovBNegativeWord dw -1206

; ShortInt format

00000013 0000000C SutulovAPositiveShortInt dd 12

00000017 FFFFFFF4 SutulovANegativeShortInt dd -12

0000001B 000004B6 SutulovBPositiveShortInt dd 1206

0000001F FFFFFB4A SutulovBNegativeShortInt dd -1206

00000023 00B80D34 SutulovCPositiveShortInt dd 12062004

00000027 FF47F2CC SutulovCNegativeShortInt dd -12062004

; LongInt format

0000002B SutulovAPositiveLongInt dq 12

000000000000000C

00000033 SutulovANegativeLongInt dq -12

FFFFFFFFFFFFFFF4

0000003B SutulovBPositiveLongInt dq 1206

00000000000004B6

00000043 SutulovBNegativeLongInt dq -1206

FFFFFFFFFFFFFB4A

0000004B SutulovCPositiveLongInt dq 12062004

0000000000B80D34

00000053 SutulovCNegativeLongInt dq -12062004

FFFFFFFFFF47F2CC

; Single (float) format

0000005B 3C23D70A SutulovDPositiveSingle dd 0.010

0000005F BC23D70A SutulovDNegativeSingle dd -0.010

; Double (double) format

00000063 SutulovDPositiveDouble dq 0.010

3F847AE147AE147B

0000006B SutulovDNegativeDouble dq -0.010

BF847AE147AE147B

00000073 SutulovEPositiveDouble dq 0.981

3FEF645A1CAC0831

0000007B SutulovENegativeDouble dq -0.981

BFEF645A1CAC0831

00000083 SutulovFPositiveDouble dq 9814.487

40C32B3E56041893

0000008B SutulovFNegativeDouble dq -9814.487

C0C32B3E56041893

; Extended (long double) format

00000093 SutulovFPositiveExtended dt 9814.487

400C9959F2B020C49BA6

0000009D SutulovFNegativeExtended dt -9814.487

C00C9959F2B020C49BA6

; Window Caption

000000A7 41 73 73 65 6D SutulovWindowCaption db "Assembly lab1 Nikita Sutulov IM-12 Group, variant 20", 0

62 6C 79 20

6C 61 62 31

20 4E 69 6B

69 74 61 20

53 75 74 75

6C 6F 76 20

49 4D 2D 31

32 20 47 72

6F 75 70 2C

20 76 61 72

69 61 6E 74

20 32 30 00

; Forms

000000DC 44 61 74 65 20 SutulovDateOfBirthForm db "Date of birth: %s", 0

6F 66 20 62

69 72 74 68

3A 20 25 73

00

000000EE 41 20 3D 20 25 SutulovAForm db "A = %d", 9, 9, "-A = %d", 0

64 09 09 2D

41 20 3D 20

25 64 00

000000FE 42 20 3D 20 25 SutulovBForm db "B = %d", 9, 9, "-B = %d", 0

64 09 09 2D

42 20 3D 20

25 64 00

0000010E 43 20 3D 20 25 SutulovCForm db "C = %d", 9, "-C = %d", 0

64 09 2D 43

20 3D 20 25

64 00

0000011D 44 20 3D 20 25 SutulovDForm db "D = %s", 9, 9, "-D = %s", 0

73 09 09 2D

44 20 3D 20

25 73 00

0000012D 45 20 3D 20 25 SutulovEForm db "E = %s", 9, 9, "-E = %s", 0

73 09 09 2D

45 20 3D 20

25 73 00

0000013D 46 20 3D 20 25 SutulovFForm db "F = %s", 9, "-F = %s", 0

73 09 2D 46

20 3D 20 25

73 00

0000014C 25 73 0A 0A 41 SutulovFinalForm db "%s", 10, 10,

20 6E 75 6D

62 65 72 73

20 28 64 61

79 20 6F 66

20 62 69 72

74 68 29 3A

0A 25 73 0A

0A 42 20 6E

75 6D 62 65

72 73 20 28

64 61 79 20

61 6E 64 20

6D 6F 6E 74

68 20 6F 66

20 62 69 72

74 68 29 3A

0A 25 73 0A

0A 43 20 6E

75 6D 62 65

72 73 20 28

66 75 6C 6C

20 64 61 74

65 20 6F 66

20 62 69 72

74 68 29 3A

0A 25 73 0A

0A 44 20 6E

75 6D 62 65

72 73 20 28

41 20 64 69

76 69 64 65

64 20 62 79

20 74 68 65

20 67 72 61

64 65 62 6F

6F 6B 20 6E

75 6D 62 65

72 29 3A 0A

25 73 0A 0A

45 20 6E 75

6D 62 65 72

73 20 28 42

20 64 69 76

69 64 65 64

20 62 79 20

74 68 65 20

67 72 61 64

65 62 6F 6F

6B 20 6E 75

6D 62 65 72

29 3A 0A 25

73 0A 0A 46

20 6E 75 6D

62 65 72 73

20 28 43 20

64 69 76 69

64 65 64 20

62 79 20 74

68 65 20 67

72 61 64 65

62 6F 6F 6B

20 6E 75 6D

62 65 72 29

3A 0A 25 73

0A 00

"A numbers (day of birth):", 10, "%s", 10, 10,

"B numbers (day and month of birth):", 10, "%s", 10, 10,

"C numbers (full date of birth):", 10, "%s", 10, 10,

"D numbers (A divided by the gradebook number):", 10, "%s", 10, 10,

"E numbers (B divided by the gradebook number):", 10, "%s", 10, 10,

"F numbers (C divided by the gradebook number):", 10, "%s", 10, 0

00000000 .data?

00000000 00000080 [ SutulovDateOfBirthBuffer db 128 dup (?)

00

]

00000080 00000040 [ SutulovABuffer db 64 dup (?)

00

]

000000C0 00000040 [ SutulovBBuffer db 64 dup (?)

00

]

00000100 00000040 [ SutulovCBuffer db 64 dup (?)

00

]

00000140 00000040 [ SutulovDBuffer db 64 dup (?)

00

]

00000180 00000040 [ SutulovEBuffer db 64 dup (?)

00

]

000001C0 00000040 [ SutulovFBuffer db 64 dup (?)

00

]

00000200 00000020 [ SutulovDPositiveStringBuffer db 32 dup (?)

00

]

00000220 00000020 [ SutulovDNegativeStringBuffer db 32 dup (?)

00

]

00000240 00000020 [ SutulovEPositiveStringBuffer db 32 dup (?)

00

]

00000260 00000080 [ SutulovENegativeStringBuffer db 128 dup (?)

00

]

000002E0 00000080 [ SutulovFPositiveStringBuffer db 128 dup (?)

00

]

00000360 00000080 [ SutulovFNegativeStringBuffer db 128 dup (?)

00

]

000003E0 00000400 [ SutulovFinalBuffer db 1024 dup (?)

00

]

00000000 .code

00000000 start:

invoke FloatToStr2, SutulovDPositiveDouble, offset SutulovDPositiveStringBuffer

invoke FloatToStr2, SutulovDNegativeDouble, offset SutulovDNegativeStringBuffer

invoke FloatToStr2, SutulovEPositiveDouble, offset SutulovEPositiveStringBuffer

invoke FloatToStr2, SutulovENegativeDouble, offset SutulovENegativeStringBuffer

invoke FloatToStr2, SutulovFPositiveDouble, offset SutulovFPositiveStringBuffer

invoke FloatToStr2, SutulovFNegativeDouble, offset SutulovFNegativeStringBuffer

invoke wsprintf, offset SutulovDateOfBirthBuffer, offset SutulovDateOfBirthForm, offset SutulovDateOfBirthCharacterString

invoke wsprintf, offset SutulovABuffer, offset SutulovAForm, SutulovAPositiveShortInt, SutulovANegativeShortInt

invoke wsprintf, offset SutulovBBuffer, offset SutulovBForm, SutulovBPositiveShortInt, SutulovBNegativeShortInt

invoke wsprintf, offset SutulovCBuffer, offset SutulovCForm, SutulovCPositiveShortInt, SutulovCNegativeShortInt

invoke wsprintf, offset SutulovDBuffer, offset SutulovDForm, offset SutulovDPositiveStringBuffer, offset SutulovDNegativeStringBuffer

invoke wsprintf, offset SutulovEBuffer, offset SutulovEForm, offset SutulovEPositiveStringBuffer, offset SutulovENegativeStringBuffer

invoke wsprintf, offset SutulovFBuffer, offset SutulovFForm, offset SutulovFPositiveStringBuffer, offset SutulovFNegativeStringBuffer

invoke wsprintf, offset SutulovFinalBuffer, offset SutulovFinalForm,

offset SutulovDateOfBirthBuffer,

offset SutulovABuffer,

offset SutulovBBuffer,

offset SutulovCBuffer,

offset SutulovDBuffer,

offset SutulovEBuffer,

offset SutulovFBuffer

invoke MessageBox, 0, offset SutulovFinalBuffer, offset SutulovWindowCaption, 0

invoke ExitProcess, 0

end start

**8. Порівняти результати, що містяться в лістингу, з розрахунковими, отриманими при виконанні пунктів 3 – 6:**

| Константа | Дані зі звіту | Дані з лістингу | Чи співпадають дані |
| --- | --- | --- | --- |
| “ddmmyyyy” | “1” = 3116  “2” = 3216  “0” = 3016  “6” = 3616  “2” = 3216  “0” = 3016  “0” = 3016  “4” = 3416 | 31 32 30 36 32 30 30 34 00 | Так |
| A (однобайтовий формат) | 0С16 | 0C SutulovAPositiveByte db 12 | Так |
| -A (однобайтовий формат) | F416 | F4 SutulovANegativeByte db -12 | Так |
| A (двобайтовий формат) | 00 0C16 | 000C SutulovAPositiveWord dw 12 | Так |
| -A (двобайтовий формат) | FF F416 | FFF4 SutulovANegativeWord dw -12 | Так |
| A (чотирьохбайтовий формат) | 00 00 00 0C16 | 0000000C SutulovAPositiveShortInt dd 12 | Так |
| -A (чотирьохбайтовий формат) | FF FF FF F416 | FFFFFFF4 SutulovANegativeShortInt dd -12 | Так |
| A (восьмибайтовий формат) | 00 00 00 00 00 00 00 0C16 | SutulovAPositiveLongInt dq 12  000000000000000C | Так |
| -A (восьмибайтовий формат) | FF FF FF FF FF FF FF F416 | SutulovANegativeLongInt dq -12  FFFFFFFFFFFFFFF4 | Так |
| B (двобайтовий формат) | 04 B616 | 04B6 SutulovBPositiveWord dw 1206 | Так |
| -B (двобайтовий формат) | FB 4A16 | FB4A SutulovBNegativeWord dw -1206 | Так |
| B (чотирьохбайтовий формат) | 00 00 04 B616 | 000004B6 SutulovBPositiveShortInt dd 1206 | Так |
| -B (чотирьохбайтовий формат) | FF FF FB 4A16 | FFFFFB4A SutulovBNegativeShortInt dd -1206 | Так |
| B (восьмибайтовий формат) | 00 00 00 00 00 00 04 B616 | SutulovBPositiveLongInt dq 1206  00000000000004B6 | Так |
| -B (восьмибайтовий формат) | FF FF FF FF FF FF FB 4A16 | SutulovBNegativeLongInt dq -1206  FFFFFFFFFFFFFB4A | Так |
| C (чотирьохбайтовий формат) | 00 B8 0D 3416 | 00B80D34 SutulovCPositiveShortInt dd 12062004 | Так |
| -C (чотирьохбайтовий формат) | FF 47 F2 CC16 | FF47F2CC SutulovCNegativeShortInt dd -12062004 | Так |
| C (восьмибайтовий формат) | 00 00 00 00 00 B8 0D 3416 | SutulovCPositiveLongInt dq 12062004  0000000000B80D34 | Так |
| -C (восьмибайтовий формат) | FF FF FF FF FF 47 F2 CC16 | SutulovCNegativeLongInt dq -12062004  FFFFFFFFFF47F2CC | Так |

| D (Чотирьохбайтовий формат Single(float)) | 3C 20 00 0016 | 3C23D70A SutulovDPositiveSingle dd 0.010 | Співпадають перші три цифри, бо ми округлювали до 10 знаків після коми |
| --- | --- | --- | --- |
| -D  (Чотирьохбайтовий формат Single(float)) | BC 20 00 0016 | BC23D70A SutulovDNegativeSingle dd -0.010 | Співпадають перші три цифри, бо ми округлювали до 10 знаків після коми |
| E  (Восьмибайтовий формат Double(double)) | 3F EF 68 00 00 00 00 0016 | SutulovEPositiveDouble dq 0.981  3FEF645A1CAC0831 | Співпадають перші п’ять цифр, бо ми округлювали до 10 знаків після коми |
| -E  (Восьмибайтовий формат Double(double)) | BF EF 68 00 00 00 00 0016 | SutulovENegativeDouble dq -0.981  BFEF645A1CAC0831 | Співпадають перші п’ять цифр, бо ми округлювали до 10 знаків після коми |
| F (Десятибайтовий формат Extended (long double)) | 40 0C 99 59 F3 00 00 00 00 0016 | SutulovFPositiveExtended dt 9814.487  400C9959F2B020C49BA6 | Співпадають перші дев’ять цифр, бо ми округлювали до 10 знаків після коми |
| -F  (Десятибайтовий формат Extended (long double)) | C0 0C 99 59 F3 00 00 00 00 0016 | SutulovFNegativeExtended dt -9814.487  C00C9959F2B020C49BA6 | Співпадають перші дев’ять цифр, бо ми округлювали до 10 знаків після коми |

**9. Зробити висновки по лабораторній роботі.**

В результаті виконання цієї лабораторної роботи я вивчив формати внутрішнього представлення цілих і дійсних чисел, засвоїв навички написання базових програм на мові Assembler, а також порівняв результати, отримані теоретично, з результатами, що містяться в отриманому розширеному лістингу програми. Після порівняння я дійшов до висновку, що я перевів всі числа правильно, проте також вважаю за необхідне відмітити, що точність теоретично розрахованих дробових чисел залежить від кількості знаків після коми, на які ми округлюємо їх при переведенні в двійкову систему счислення.