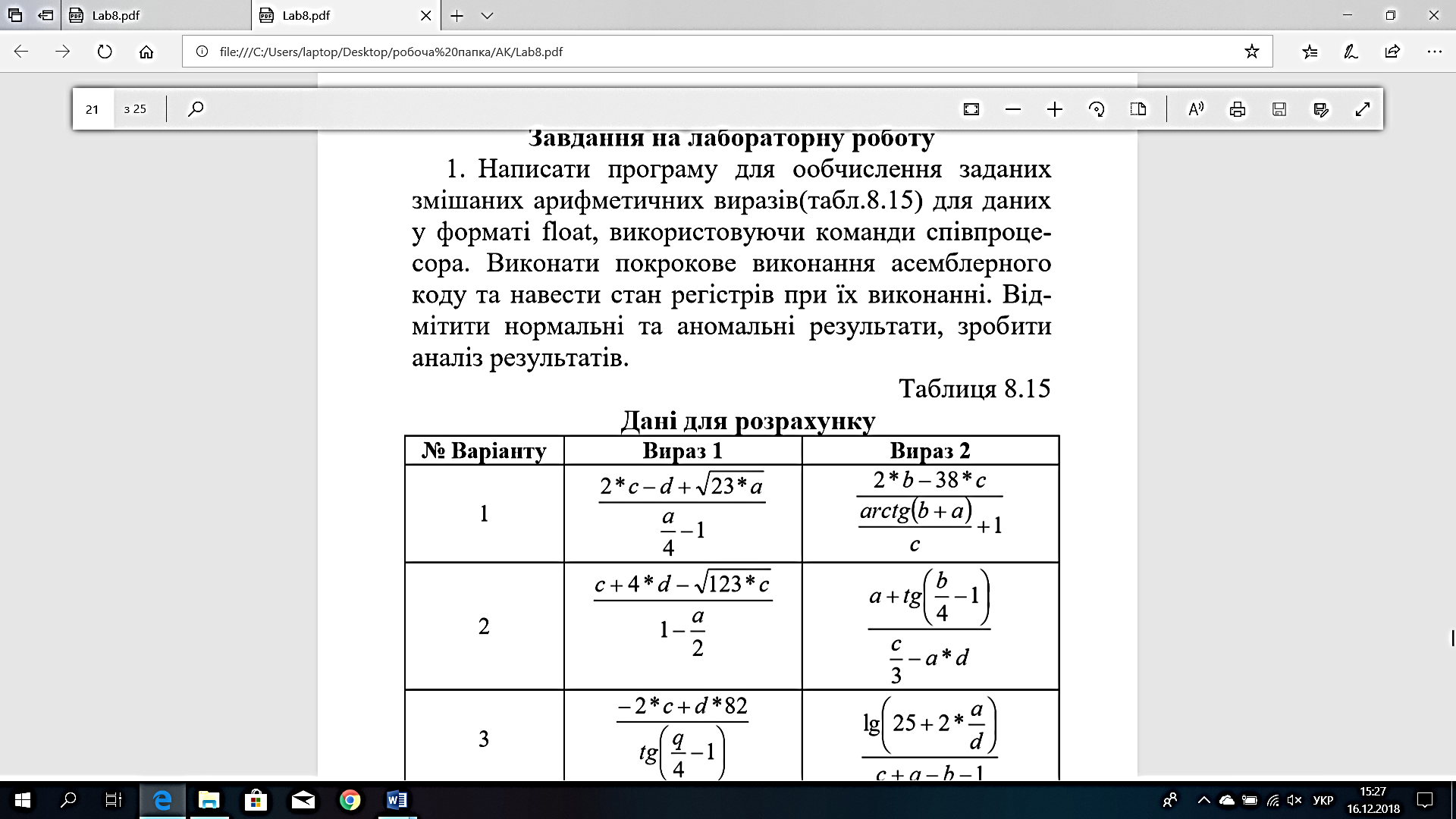
**ЛАБОРАТОРНА РОБОТА № 8**

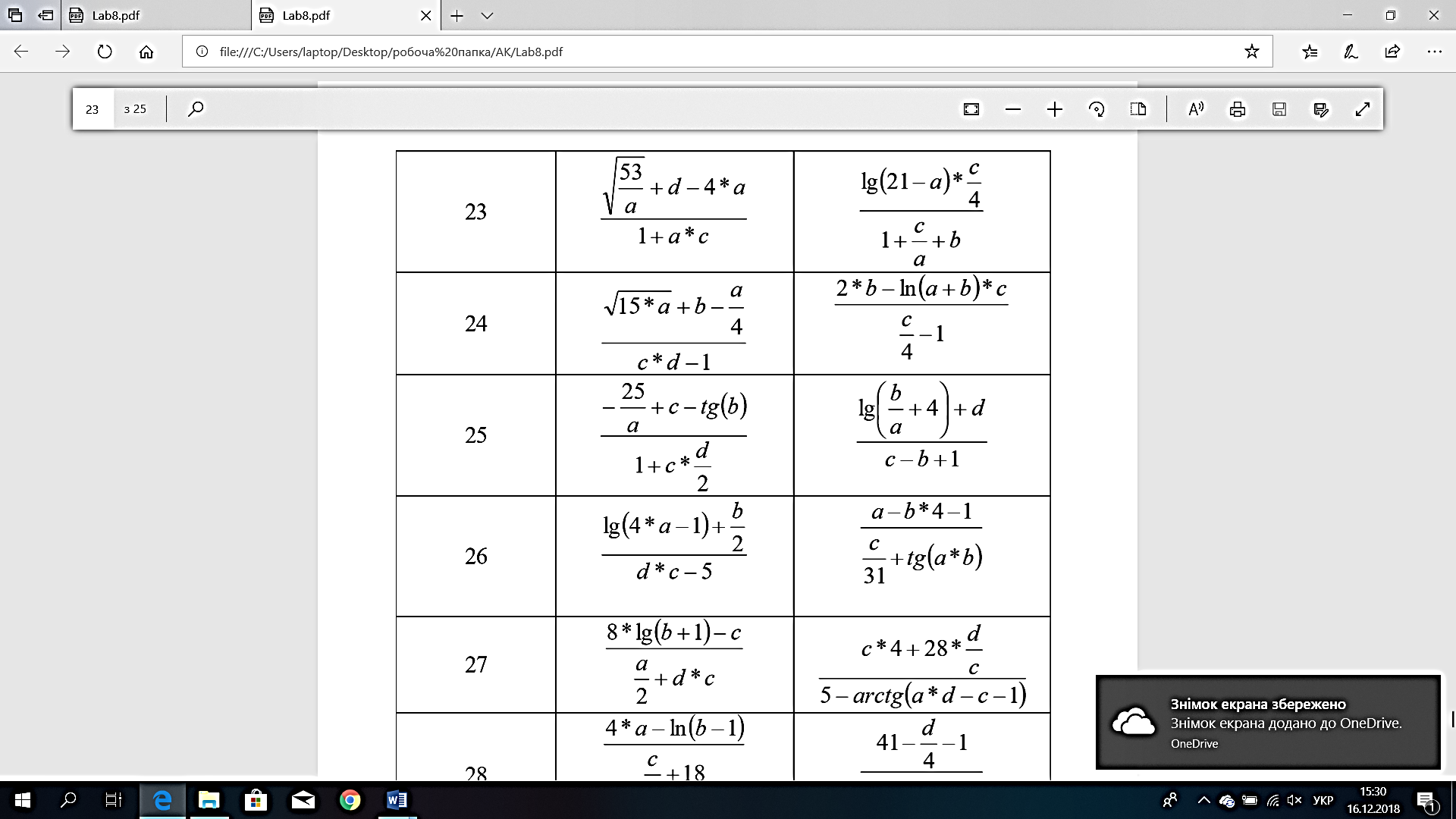
**ОБЧИСЛЕННЯ АРИФМЕТИЧНИХ ВИРАЗІВ І ТРАНСЦЕНДЕНТНИХ ФУНКЦІЙ (СПІВПРОЦЕСОР іх87)**

***Мета:*** ознайомитися з основними командами мови Assembler для обчислення арифметичних виразів і трансцендентних функцій; набути практичних навичок в написанні програм, які обчислюють заданий змішаний арифметичний вираз, використовуючи арифметичні операції співпроцесора на мові Assembler.

**Хід роботи:**

**Завдання 1**: Написати програму для ообчислення заданих змішаних арифметичних виразів для даних у форматі float, використовуючи команди співпроцесора. Виконати покрокове виконання асемблерного коду та навести стан регістрів при їх виконанні. Відмітити нормальні та аномальні результати, зробити аналіз результатів.





**Вираз 1.**

Лістинг програми:

#include "pch.h"

#include <iostream>

#include <time.h>

#include <Windows.h>

#include <math.h>

#include <conio.h>

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

float a, b, c, d, res\_c, res\_asm, m = 15, n = 4, k = 1;

do {

printf("Введіть значення:\n");

printf("a = "), scanf\_s("%f", &a);

printf("b = "), scanf\_s("%f", &b);

printf("c = "), scanf\_s("%f", &c);

printf("d = "), scanf\_s("%f", &d);

} while (a > 2147483647.0 || a < -2147483648.0 || b > 2147483647.0 || b < -2147483648.0 || c > 2147483647.0 || c < -2147483648.0 || d > 2147483647.0 || d < -2147483648.0);

if ((c\*d - 1) == 0) {

printf("Помилка! Ділення на 0.\n");

}

else {

res\_c = (sqrt(15 \* a) + b - a / 4) / (c\*d - 1);

printf("Результат на мові C: %f\n", res\_c);

}

\_asm

{

finit // ініціалізація співпроцесора

fld a // завантаження в вершину стека

fmul m // множення (15\*a)

fsqrt // квадратний корінь sqrt(15\*a)

fadd b // додавання sqrt(15\*a)+b

fld a // завантаження в вершину стека

fdiv n // ділення (a/4)

fsub // віднімання sqrt(15\*a)+b-a/4

fld c // завантаження в вершину стека

fmul d // множення (c\*d)

fsub k // віднімання (c\*d-1)

fdiv // ділення (sqrt(15\*a)+b-a/4)/(c\*d-1)

fstp res\_asm // res\_asm = (sqrt(15\*a)+b-a/4)/(c\*d-1)

}

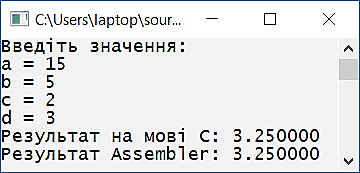
printf("Результат Assembler: %f\n", res\_asm);

\_getch();

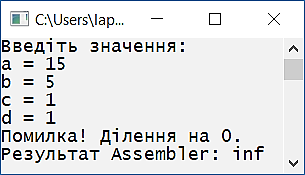
return 0;

}

Результат виконання програми:



Перевіримо виникнення помилки при діленні на 0



Виникнення помилки при діленні на 0

**Значення регістрів при покроковому виконанні**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Команда** | **Опис** | **Стан регістрів** | | **Коментар** |
| finit | Ініціалізація співпроцесора | cwr=037Fh; swr=0; twr=FFFFh; dpr=0; ipr=0; | |  |
| fld a | Завантаження в вершину стека | src | | a |
| fmul m | Множення | dst=dst\*src | | a\*15 |
| fsqrt | Квадратний корінь | St(0)=√𝑆𝑡(0) | | sqrt(15\*a) |
| fadd b | Додавання | dst=dst+src | | sqrt(15\*a)+b |
| fld a | Завантаження в вершину стека | src | | a |
| fdiv n | Ділення | | dst=dst/src | a/4 |
| fsub | Віднімання | | dst=dst-src | sqrt(15\*a)+b-a/4 |
| fld c | Завантаження в вершину стека | | src | c |
| fmul d | Множення | | dst=dst\*src | c\*d |
| fsub k | Віднімання | | dst=dst-src | c\*d-1 |
| fdiv | Ділення | | dst=dst/src | (sqrt(15\*a)+b-a/4)/(c\*d-1) |
| fstp res\_asm | Збереження у вершині стеку | | dst= St(0) | (sqrt(15\*a)+b-a/4)/(c\*d-1) |

**Вираз 2.**

Лістинг програми:

#include "pch.h"

#include <iostream>

#include <time.h>

#include <Windows.h>

#include <math.h>

#include <conio.h>

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

float a, b, c, d, res\_c, res\_asm, m = 2, n = 4, k = 1;

do {

printf("Введіть значення:\n");

printf("a = "), scanf\_s("%f", &a);

printf("b = "), scanf\_s("%f", &b);

printf("c = "), scanf\_s("%f", &c);

} while (a > 2147483647.0 || a < -2147483648.0 || b > 2147483647.0 || b < -2147483648.0 || c > 2147483647.0 || c < -2147483648.0);

if ((c / 4 - 1) == 0) {

printf("Помилка! Ділення на 0.\n");

}

else {

res\_c = (2 \* b - log(a + b)\*c) / (c / 4 - 1);

printf("Результат на мові C: %f\n", res\_c);

}

\_asm

{

finit // ініціалізація співпроцесора

fld b // завантаження в вершину стека

fmul m // множення (b\*2)

fldln2 // рахуємо ln2

fld a // завантаження в вершину стека

fadd b // додавання (a+b)

fyl2x // логарифм ln(a+b)

fmul c // множення ln(a+b)\*c

fsub // віднімання 2\*b-ln(a+b)\*c

fld c // завантаження в вершину стека

fdiv n // ділення (c/4)

fsub k // віднімання (c/4-1)

fdiv // ділення (2\*b-ln(a+b)\*c)/(c/4-1)

fstp res\_asm // res\_asm=(2\*b-ln(a+b)\*c)/(c/4-1)

}

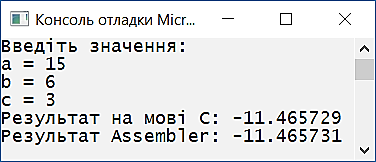
printf("Результат Assembler: %f\n", res\_asm);

\_getch();

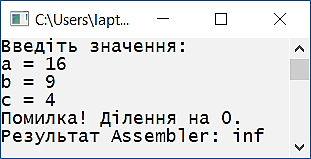
return 0;

}

Результат виконання програми:



Перевіримо виникнення помилки при діленні на 0



Виникнення помилки при діленні на 0

**Значення регістрів при покроковому виконанні**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Команда** | **Опис** | **Стан регістрів** | | **Коментар** |
| finit | Ініціалізація співпроцесора | cwr=037Fh; swr=0; twr=FFFFh; dpr=0; ipr=0; | |  |
| fld b | Завантаження в вершину стека | src | | b |
| fmul m | Множення | dst=dst\*src | | b\*2 |
| fldln2 | Pахуємо ln2 | dst=ln 2 | |  |
| fld a | Завантаження в вершину стека | src | | a |
| fadd b | Додавання | dst=dst+src | | a+b |
| fyl2x | Логарифм | St(0)=ln(𝑆𝑡(0)) | | ln(a+b) |
| fmul c | Множення | | dst=dst\*src | ln(a+b)\*c |
| fsub | Віднімання | | dst=dst-src | 2\*b-ln(a+b)\*c |
| fld c | Завантаження в вершину стека | | src | c |
| fdiv n | Ділення | | dst=dst/src | c/4 |
| fsub k | Віднімання | | dst=dst-src | c/4-1 |
| fdiv | Ділення | | dst=dst/src | (2\*b-ln(a+b)\*c) / ( c/4-1) |
| fstp res\_asm | Збереження у вершині стеку | | dst= St(0) | (2\*b-ln(a+b)\*c) / ( c/4-1) |

***Висновок:*** в ході виконання лабораторної роботи ознайомлено з основними командами мови Assembler для обчислення арифметичних виразів і трансцендентних функцій; набутo практичних навичок в написанні програм, які обчислюють заданий змішаний арифметичний вираз, використовуючи арифметичні операції співпроцесора на мові Assembler.