МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Вятский государственный университет»

Институт математики и информационных систем

Факультет автоматики и вычислительной техники

Кафедра радиоэлектронных средств

Отчет по лабораторной работе №4

по дисциплине «Цифровые устройства и микропроцессоры»

Вариант 12

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Выполнил: | студент группы  ИНБб-3301-02-00 |  |  | А.В. Кригер |
|  |  |  |  |  |
| Проверил: | преподаватель |  |  | М.А. Земцов |
|  |  |  |  |  |

Работа защищена с оценкой «\_\_\_\_\_\_\_\_\_\_\_\_\_» «\_\_\_»\_\_\_\_\_\_\_ 2025 г.

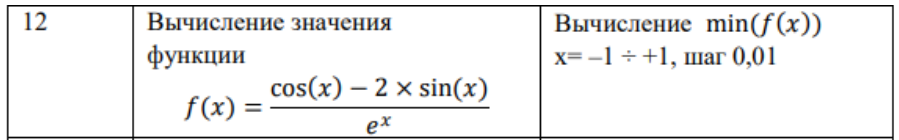
г. Киров

2025

**Цель работы:** знакомство с технологией применения языка ассемблера

при разработке программного обеспечения на языках высокого уровня.

**Задание:**



**Ход работы:**

Код программы с комментариями:

Блок кода Sourse.cpp

1.#include <iostream>

2.#include <cmath>

3.using namespace std;

4.extern "C" float FindMin(float iStep);

5.int main(int argc, char\*\* argv)

6.{

7. float iStep = 0.01; // Начальное значение шага цикла

8. cout << "Input step: ";

9. cin >> iStep;

10. cout << "\nf(x):\n";

11. float Res = 0.0;

12. Res = FindMin(iStep); // Вызов ассемблерной функции FindMin для нахождения минимального значения

13. cout << "End!\nmin(f(x)) = " << Res << endl; // Вывод результата на консоль

14. return 0;

15.}

16.extern "C" float fun(float x)

17.{

18. float f;

19. // Вычисление значения функции: f(x) = (cos(x) - 2\*sin(x)) / exp(x\*x)

20. f = (cos(x) - (2 \* sin(x))) / exp(x \* x);

21. cout << f << endl;

22. return f;

23.}

Блок кода Sourse1.asm

1.586

2.MODEL flat, C

3.DATA

4.i REAL4 -1.0 ; Начальное значение i цикла

5.iEnd REAL4 2.0 ; Конечное значение i цикла

6.iStep REAL4 0.6 ; Шаг цикла

7.Min REAL4 9999.0 ; Минимальное число (начальное значение - очень большое)

8.code

9.extern fun:near

10.public FindMin

11.FindMin proc C

12.; Выгружаем из стека переданное функцией число (Шаг цикла)

13.push ebp

14.mov ebp, esp

15.mov ecx, dword ptr [ebp+8]

16.mov iStep, ecx

17.for\_start:

18. ; Проверка цикла, если i > iEnd выход из цикла

19. fld iEnd

20. fld i

21. fcomip st(0), st(1)

22. fstp st(0)

23. ja for\_end

24. ; Тело цикла:

25. push ecx ; Сохраняем ecx перед вызовом функции

26. push i ; Передаем i как параметр

27. call fun ; Вызов внешней функции fun(i)

28. add esp, 4 ; Очистка стека после вызова функции

29. ; Сравниваем результат fun(i) с текущим минимальным значением

30. fld Min ; Загружаем текущее минимальное значение

31. fcomip st(0), st(1) ; Сравниваем Min с результатом fun(i)

32. jae update\_min ; Перейти к обновлению минимума, если Min >= f(i)

33. jmp next\_iteration ; Если Min < f(i), перейти к следующей итерации

34.update\_min:

35. fstp Min ; Обновляем Min значением f(i)

36.next\_iteration:

37. fstp st(0) ; Очистка st(0) после сравнения

38. ; Прибавление к i шага цикла

39. fld i ; Загружаем текущее значение i

40. fld iStep ; Загружаем шаг цикла

41. fadd st(0), st(1) ; i = i + iStep

42. fstp i ; Сохраняем новое значение i

43. fstp st(0) ; Очистка стека

44. jmp for\_start ; Переход к следующей итерации цикла

45.for\_end:

46. fld Min ; Загружаем найденное минимальное значение

47. mov esp, ebp ; Восстанавливаем указатель стека

48. pop ebp ; Восстанавливаем базовый указатель стека

49. ret ; Возврат из функции

50.FindMin endp

51.End

Скриншоты работы программы представлены ниже на рисунках 1-4.

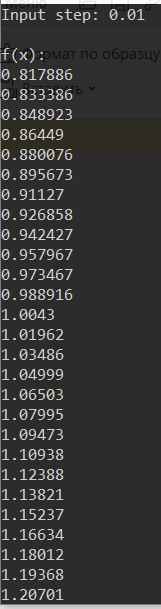


Рисунок 1 – ввод шага и работа программы

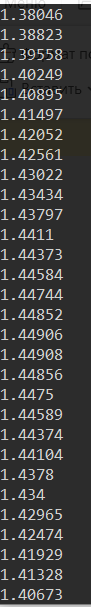


Рисунок 2 – вывод результата

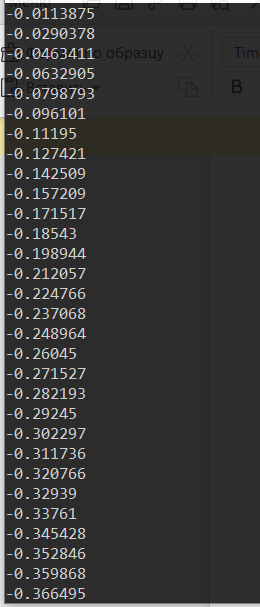


Рисунок 3 – вывод результата

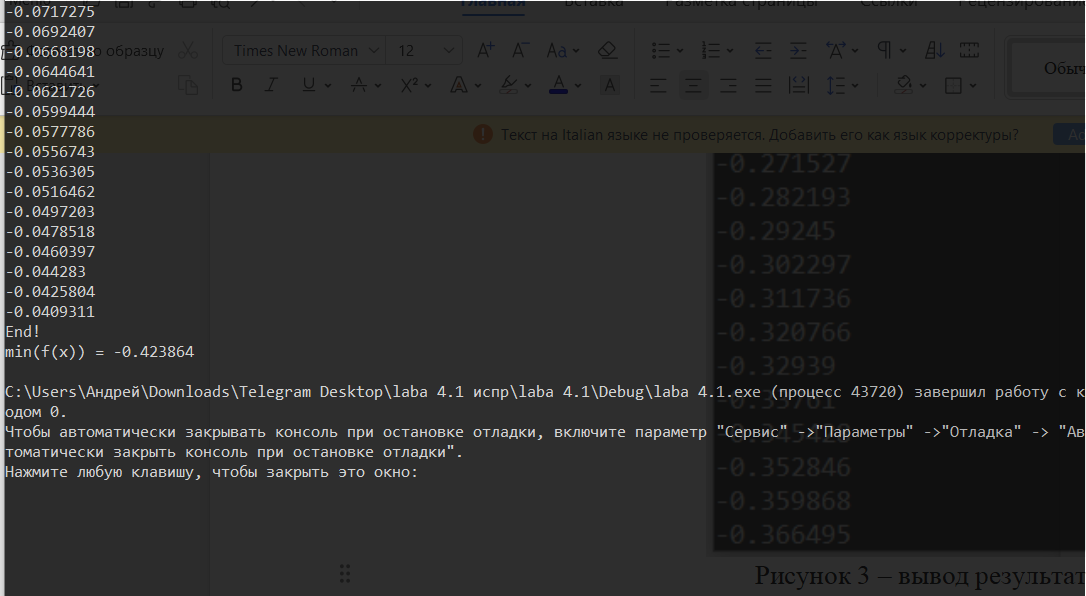


Рисунок 4 – вывод результата и ответ

**Вывод:** Выполнив данную лабораторную работу мы ознакомились с технологией применения языка ассемблера при разработке программного обеспечения на языках высокого уровня.