Министерство образования и науки РФ

Пермский национальный исследовательский политехнический университет

Электротехнический факультет

Кафедра Информационные технологии и автоматизированные системы

Организация ЭВМ и Систем

Лабораторная работа № 4

Тема: «Программирование разветвляющегося процесса»

Вариант №7

Выполнили студенты группы РИС-23-3б:

Комягин Данил Александрович

Гордеев Василий Андреевич

Проверил: доцент кафедры ИТАС

Погудин Андрей Леонидович

Пермь 2025

ВВЕДЕНИЕ

Цель работы – ознакомление с использованием команд условной передачи управления для реализации алгоритмов, пути в которых зависят от исходных данных.

Команды машинной программы выполняются в том порядке, в котором они записаны в памяти. Но время от времени этот естественный порядок приходится нарушать с тем, чтобы следующей выполнялась не очередная команда программы, а какая-то иная. Такую возможность обеспечивают команды перехода (передачи управления).

Переходы бывают условными и безусловными. Если переход делается только тогда, когда выполнено некоторое условие, то такой переход называется условным, а если он делается независимо от каких-либо условий, то это безусловный переход.

Наиболее простыми способами адресации операндов в машинных командах являются непосредственная и прямая адресации.

При непосредственной адресации в адресном поле команды вместо адреса содержится непосредственно сам операнд. Этот способ может применяться при выполнении арифметических операций, операций сравнения, а также для загрузки констант в регистры.

Достоинствами непосредственной адресации являются:

1) сокращение времени выполнения команды, так как не требуется обращения к памяти за операндом;

2) экономия памяти, поскольку отпадает необходимость в ячейке для хранения операнда.

Можно отметить следующие недостатки данного способа:

1) в адресном поле могут быть указаны только константы;

2) размер непосредственного операнда ограничен длиной адресного поля команды.

В случае прямой адресации операндов адресный код прямо указывает номер ячейки памяти, к которой производится обращение.

При всей простоте использования данный способ имеет два существенных недостатка:

1) ограниченный размер адресного пространства, так как для адресации к памяти большой емкости нужно «длинное» адресное поле;

2) ограниченные возможности по произвольному размещению программы в памяти, вследствие невозможности изменения адреса в процессе вычислений.

ХОД РАБОТЫ

На рисунке 1 показаны значения i, j и k для каждого из вариантов. Работа выполнена по варианту №8.

Изображение выглядит как текст, число, календарь, кроссворд

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 1 - Значения i, j и k для каждого из вариантов

На рисунке 2 представлены формулы и ограничения аргумента для каждого из вариантов.

Изображение выглядит как текст, снимок экрана, число, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 2 - Формулы и ограничения аргумента

1. Разработка программы вычисления и вывода значения функции:

Ограничения для аргумента х по обеим формулам представлены на первой числовой оси (рисунок 3).

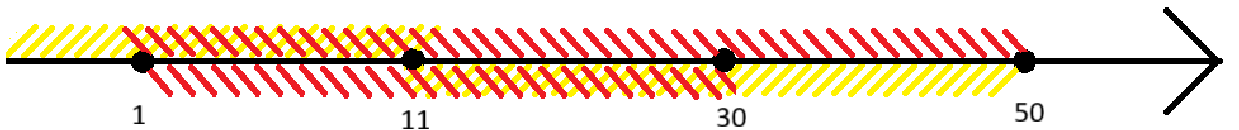


Рисунок 3 – Ограничения аргумента х по заданным формулам

Зная общие ограничения для аргумента х, можно построить ось с областью допустимых значений аргумента и формулами, которые будут использоваться в зависимости от значения х.

На недопустимые значения аргумента программа выдаёт на OR максимальное отрицательное число 199 999.

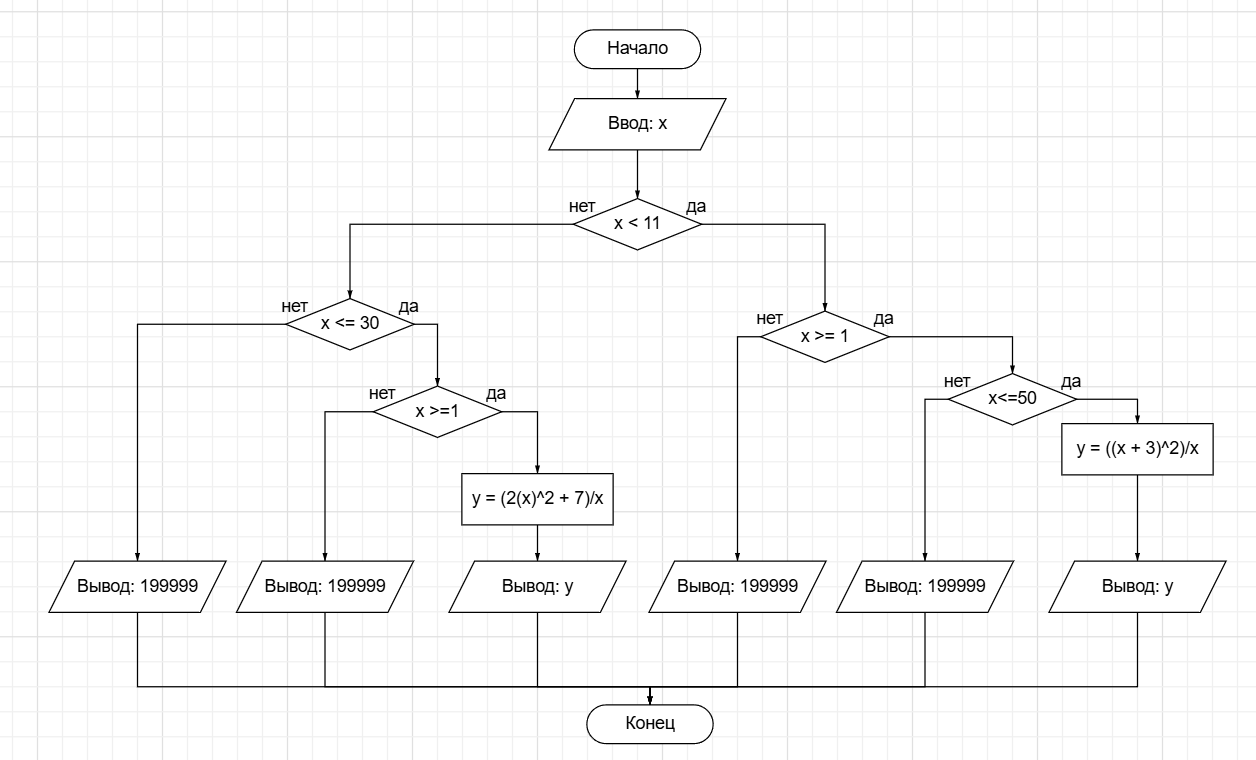


Рисунок 4 – Граф-схема алгоритма работы программы

Написанная программа с комментариями к командам приведена в таблице 1.

Таблица 1 – Описание команд программы

|  |  |
| --- | --- |
| Команда | Описание |
| IN | Занесение данных из IR в Асс (аргумент х) |
| WR 30 | Запись в ячейку 030 данных из Асс |
| SUB #11 | Вычитаем 11 из аргумента для проверки |
| JS M1 | Проверка: если х – 11 < 0, переход по метке M1 |
| RD 30 | Записываем в Асс исходный аргумент из ячейки 030 |
| SUB #31 | Вычитаем 31 из аргумента для проверки |
| JNS M3 | Проверка: если х – 31 >= 0, переход по метке M3 |
| RD 30 | Записываем в Асс исходный аргумент из ячейки 030 |
| MUL 30 | Расчёт по формуле: х \*= х |
| MUL #2 | Расчёт по формуле: х2 \*= 2 |
| ADD #7 | Расчёт по формуле: 2х2 += 7 |
| DIV 30 | Расчёт по формуле: (2х2 + 7) /= х |
| JMP M2 | Безусловный переход по метке М2 |
| M1: RD 30 | Метка М1: записываем в Асс исходный х из ячейки 030 |
| SUB #1 | Вычитаем 1 из аргумента для проверки |
| JS M3 | Проверка: если х – 1 < 0, переход по метке M3 |
| RD 30 | Записываем в Асс исходный аргумент из ячейки 030 |
| ADD #3 | Расчёт по формуле: х += 3 |
| WR 31 | Записываем в ячейку 031 значение х + 3 |
| MUL 31 | Расчёт по формуле: (х + 3) \*= (х + 3) |
| DIV 30 | Расчёт по формуле: (х + 3)2 /= х |
| JMP M2 | Безусловный переход по метке М2 |
| M3: RD #0 | Метка М3: записываем в Асс 0, Асс = 0 |
| SBI #99999 | Асс = 199999 |
| M2: OUT | Метка М2: вывод данных из Асс в OR |
| HLT | Завершение работы |

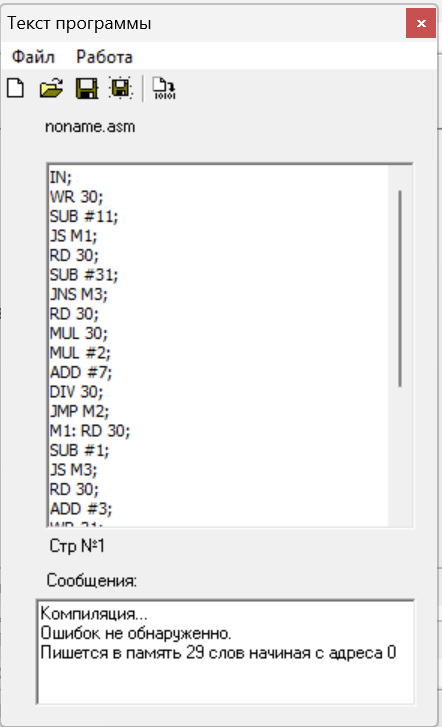
ЗАКЛЮЧЕНИЕ

В ходе лабораторной работы был успешно разработан, отлажен и протестирован алгоритм разветвляющегося вычислительного процесса для задания по варианту 7.

Программа корректно реализует вычисление функции, выбирая необходимую формулу в зависимости от значения аргумента x, и обеспечивает обработку ошибочных ситуаций выводом кода 199 999 при недопустимых значениях.

Таким образом, цель работы достигнута: приобретены практические навыки программирования разветвляющихся алгоритмов с использованием команд условной и безусловной передачи управления.

ПРИЛОЖЕНИЕ А: Компиляция программы

 Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

Приложение Б: Тестирование программы

X = 30, f(30) = 60,23

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

X = 1, f(1) = 16,00

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

X = 0, f(0) = 199999

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

X = 3, f(31) = 199999

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

X = 2, f(2) = 12,5

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.