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

Изображение выглядит как текст, число, календарь, кроссворд

Содержимое, созданное искусственным интеллектом, может быть неверным.

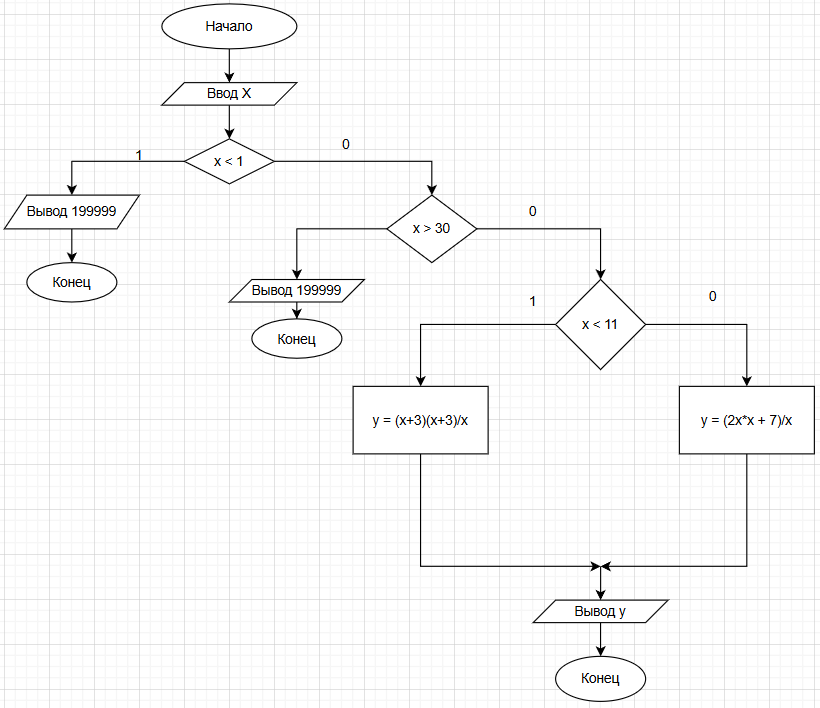
Изображение выглядит как текст, снимок экрана, число, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

1) Разработка программы вычисления и вывода значения функции:

Так, значение x не может превысить 30 или быть меньше нуля, поэтому общим интервалом для значения х выбран интервал

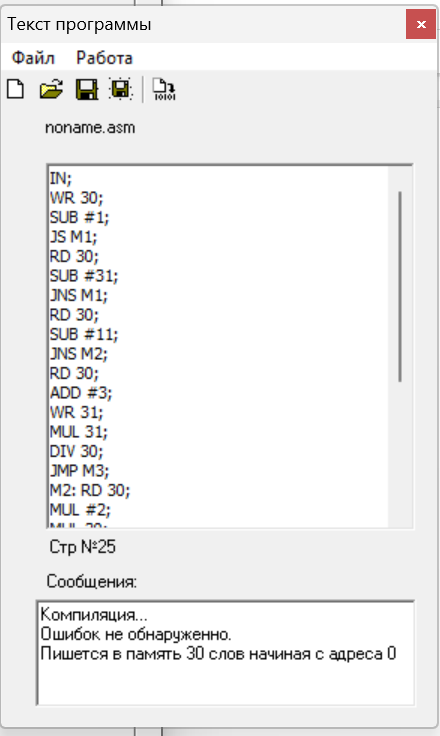
На недопустимые значения аргумента программа выдаёт на OR максимальное отрицательное число: 199 999.



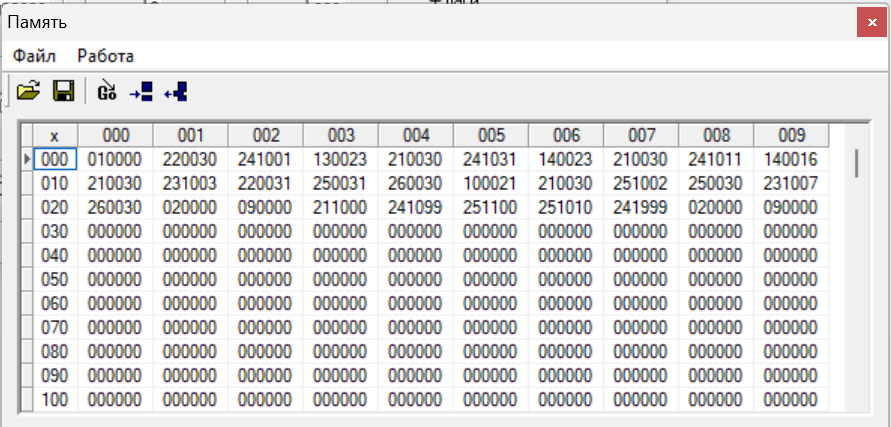
Написание программы с комментариями к командам приведено в таблице ниже:

|  |  |
| --- | --- |
| IN | Запись в Асс значения из IR |
| WR 30 | Запись в ячейку с адресом 030 значения из Асс,  т. е. в ячейке 030 будет храниться исходный Х |
| SUB #1 | Вычитание 1 из Асс (проверка на принадлежность заданному диапазону от 1 до 30) |
| JS M1 | Если число отрицательное, переходим на метку М1 |
| RD 30 | Иначе – записываем в Асс аргумент из ячейки 030 |
| SUB #31 | Вычитаем из Асс значение 30 (Проверка на принадлежность заданному диапазону от 1 до 30) |
| JNS M1 | Если число неотрицательное, переходим на метку М1 |
| RD 30 | Записываем в Асс аргумент из ячейки 030 |
| SUB #11 | Вычитаем из него 11 |
| JNS M2 | Если ответ неотрицательный, переходим на метку М2 |
| RD 30 | Записываем в Асс значение из ячейки 030 (исх. Х) |
| ADD #3 | Добавляем 3: Асс = х + 3 |
| WR 31 | Записываем х + 3 в ячейку с адресом 031 |
| MUL 31 | Домножаем Асс на значение из ячейки 031:  Асс = (х + 3)\*(х + 3) |
| DIV 30 | Делим значение в Асс на значение из ячейки 030:  Асс = (х + 3)\*(х + 3)/х; теперь формула готова |
| JMP M3 | Переходим на метку М3 без условия |
| M2: RD 30 | Метка М2: записываем в Асс исходный аргумент из ячейки 030 |
| MUL #2 | Домножаем значение в Асс на 2: Асс = х \* 2 |
| MUL 30 | Домножаем значение в Асс на значение из ячейки 030: Асс = х \* 2 \* х = 2\*х2 |
| ADD #7 | Прибавляем 7 к значению в Асс: 2\*х2 + 7 |
| DIV 30 | Делим значение в Асс на значение из ячейки 030:  Асс = (2\*х2 + 7)\х; формула готова |
| M3: OUT | Метка 3: вывод в OR значения из Асс |
| HLT | Выход |
| M1: RD #0 | Метка 1: Записываем 0 в Асс (нужно вывести 199999) |
| SUB #99 | Вычитаем 99 из Асс |
| MUL #100 | Домножаем на 100 значение в Асс |
| MUL #10 | Домножаем на 10 значение в Асс |
| SUB #999 | Вычитаем 999: Асс = 0-99\*100\*10-999=-199999 |
| OUT | Выводим число из Асс в OR |
| HLT | Завершение программы |

РАБОТА ПРОГРАММЫ

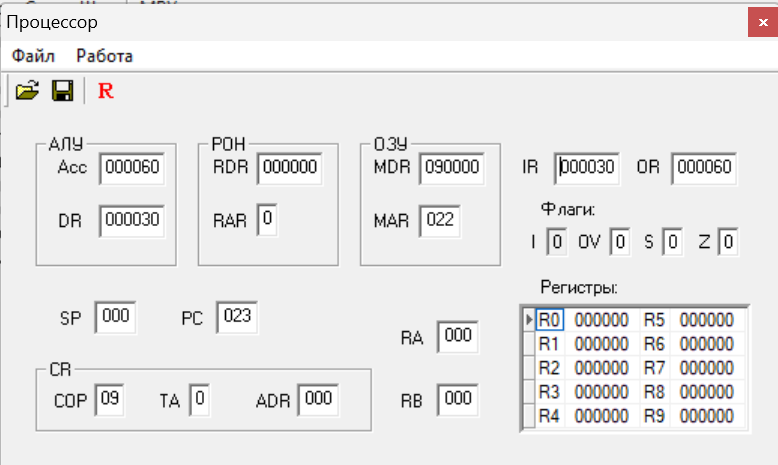
 Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.



Тестовый пример:

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



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

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

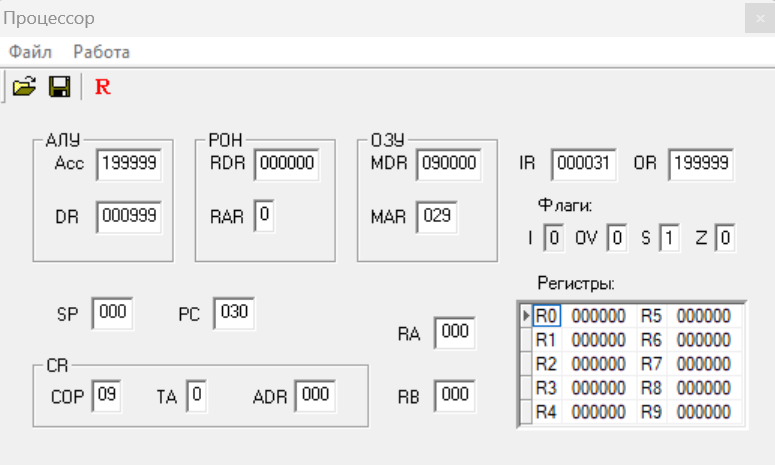
Содержимое, созданное искусственным интеллектом, может быть неверным.

X = 0 (Не принадлежит диапазону, f(0) = 199999)

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

Содержимое, созданное искусственным интеллектом, может быть неверным.

X = 31 (Не принадлежит диапазону, f(31) = 199999)



X = 2 (Не принадлежит диапазону, f(2) = 12,5)

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

Содержимое, созданное искусственным интеллектом, может быть неверным.

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.

IN

WR 30

SUB #1

JS M1

RD 30

SUB #31

JNS M1

RD 30

SUB #11

JNS M2

RD 30

ADD #3

WR 31

MUL 31

DIV 30

JMP M3

M2: RD 30

MUL #2

MUL 30

ADD #7

DIV 30

M3: OUT

HLT

M1: RD #0

SUB #99

MUL #100

MUL #10

SUB #999

OUT

HLT