

Лабораторная работа № 6

Дисциплина: Архитектура компьютера

Жибицкая Евгения Дмитриевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Задание для самостоятельной работы	11
4	Выводы	13

Список иллюстраций

2.1	Листинг 6.1	6
2.2	Запуск программы	6
2.3	Запуск измененной программы	7
2.4	Запуск lab6-2	7
2.5	Повторный запуск lab6-2	7
2.6	Запуск с iprint	8
2.7	Листинг 6.3	8
2.8	Запуск lab6-3	8
2.9	Изменение текста программы lab6-3	9
2.10	Запуск измененной программы lab6-3	9
2.11	Запуск variant	10
3.1	Написание программы	11
3.2	Запуск кода	12

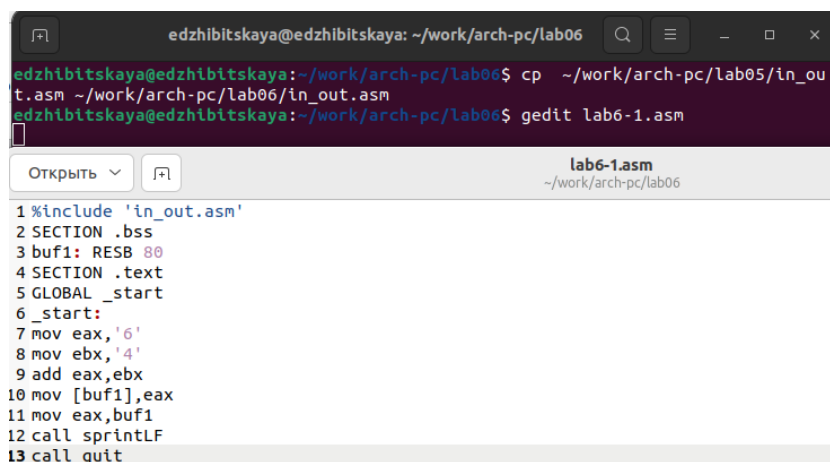
Список таблиц

1 Цель работы

Продолжение знакомства с языком ассемблера, освоение арифметических операций NASM.

2 Выполнение лабораторной работы

Начнем работу с того, что создадим каталог lab06 и файл lab6-1.asm(mkdir ~/work/arch-pc/lab06; cd ~/work/arch-pc/lab06; touch lab6-1.asm) Скопируем файл in_out.asm в lab06 и вставим код из Листинг 6.1 в lab6-1.asm(рис. 2.1).



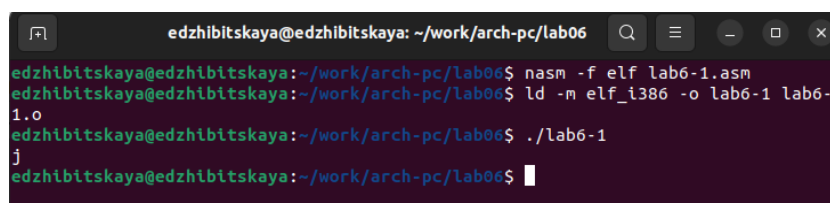
```
edzhibitskaya@edzhibitskaya: ~/work/arch-pc/lab06
edzhibitskaya@edzhibitskaya:~/work/arch-pc/lab06$ cp ~/work/arch-pc/lab05/in_out.asm ~/work/arch-pc/lab06/in_out.asm
edzhibitskaya@edzhibitskaya:~/work/arch-pc/lab06$ gedit lab6-1.asm

lab6-1.asm
~/work/arch-pc/lab06

1 %include 'in_out.asm'
2 SECTION .bss
3 buf1: RESB 80
4 SECTION .text
5 GLOBAL _start
6 _start:
7 mov eax, '6'
8 mov ebx, '4'
9 add eax, ebx
10 mov [buf1], eax
11 mov eax, buf1
12 call sprintf
13 call quit
```

Рис. 2.1: Листинг 6.1

Создадим исполняемый файл и запустим его(рис. 2.2).

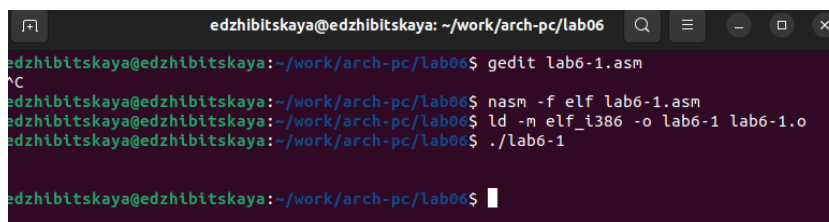


```
edzhibitskaya@edzhibitskaya: ~/work/arch-pc/lab06
edzhibitskaya@edzhibitskaya:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
edzhibitskaya@edzhibitskaya:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
edzhibitskaya@edzhibitskaya:~/work/arch-pc/lab06$ ./lab6-1
6
edzhibitskaya@edzhibitskaya:~/work/arch-pc/lab06$
```

Рис. 2.2: Запуск программы

Затем изменим текст программы и вместо символов, запишем в регистры

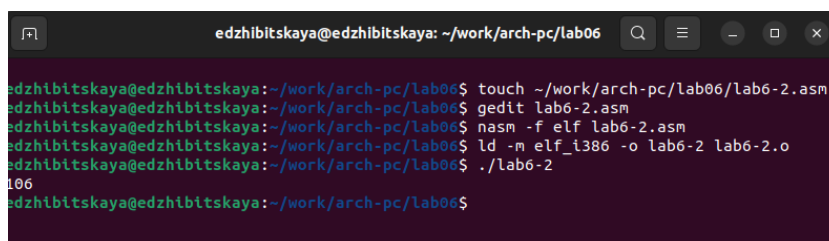
числа, так как в предыдущий раз программа не вывела ожидаемый результат, также создадим исполняемый файл и запустим программу (рис. 2.3).



```
edzhbitskaya@edzhbitskaya: ~/work/arch-pc/lab06
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab06$ gedit lab6-1.asm
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab06$ ./lab6-1
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab06$
```

Рис. 2.3: Запуск измененной программы

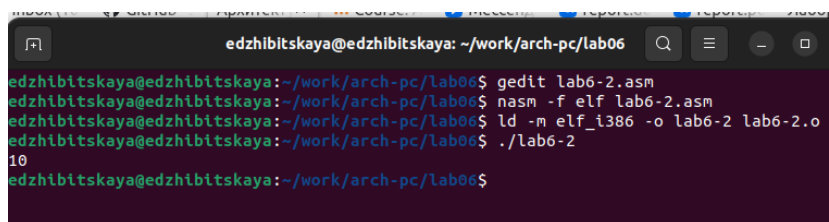
Далее создадим файл lab6-2.asm, введем в него текст из Листинг 6.2, создадим исполняемый файл и запустим(рис. 2.4).



```
edzhbitskaya@edzhbitskaya: ~/work/arch-pc/lab06
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-2.asm
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab06$ gedit lab6-2.asm
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab06$ ./lab6-2
106
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab06$
```

Рис. 2.4: Запуск lab6-2

Опять заменим символы на числа и запустим программу.(рис. 2.5).



```
edzhbitskaya@edzhbitskaya: ~/work/arch-pc/lab06
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab06$ gedit lab6-2.asm
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab06$ ./lab6-2
10
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab06$
```

Рис. 2.5: Повторный запуск lab6-2

В конце запустим программу еще раз, заменив `iprintLF` на `iprint`(отличаются наличием/отсутствием переноса строки)(рис. 2.6).

```
edzhbitskaya@edzhbitskaya: ~/work/arch-pc/lab06
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab06$ gedit lab6-2.asm
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab06$ ./lab6-2
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab06$
```

Рис. 2.6: Запуск с `iprint`

Еще создадим файл `lab6-3.asm` и заполним его текстом из Листинг 6.3(рис. 2.7).

```
edzhbitskaya@edzhbitskaya: ~/work/arch-pc/lab06
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-3.asm
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab06$ gedit lab6-3.asm
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab06$
```

Открыть ▾

lab6-3.asm
~/work/arch-pc/lab06

```
1 %include 'in_out.asm' ; подключение внешнего файла
2 SECTION .data
3 div: DB 'Результат: ',0
4 rem: DB 'Остаток от деления: ',0
5 SECTION .text
6 GLOBAL _start
7 _start:
8 ; ---- Вычисление выражения
9 mov eax,5 ; EAX=5
10 mov ebx,2 ; EBX=2
11 mul ebx ; EAX=EAX*EBX
12 add eax,3 ; EAX=EAX+3
13 xor edx,edx ; обнуляем EDX для корректной работы div
14 mov ebx,3 ; EBX=3
15 div ebx ; EAX=EAX/3, EDX=остаток от деления
16 mov edi,eax ; запись результата вычисления в 'edi'
17 ; ---- Вывод результата на экран
18 mov eax,div ; вызов подпрограммы печати
19 call sprint ; сообщения 'Результат: '
20 mov eax,edi ; вызов подпрограммы печати значения
21 call iprintLF ; из 'edi' в виде символов
22 mov eax,rem ; вызов подпрограммы печати
23 call sprint ; сообщения 'Остаток от деления: '
24 mov eax,edx ; вызов подпрограммы печати значения
25 call iprintLF ; из 'edx' (остаток) в виде символов
26 call quit ; вызов подпрограммы завершения
```

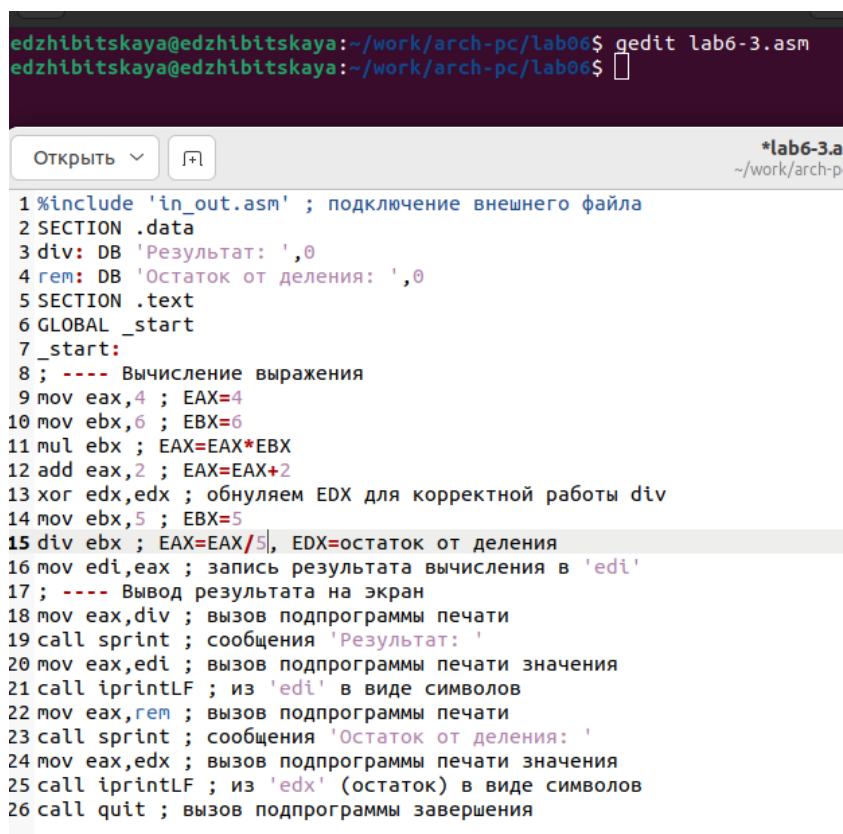
Рис. 2.7: Листинг 6.3

Создадим исполняемый файл и запустим его.(рис. 2.8).

```
edzhbitskaya@edzhbitskaya: ~/work/arch-pc/lab06
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab06$
```

Рис. 2.8: Запуск `lab6-3`

Изменим имеющуюся программу, для вычисления $f(x) = (4 * 6 + 2)/5$ (рис. 2.9).



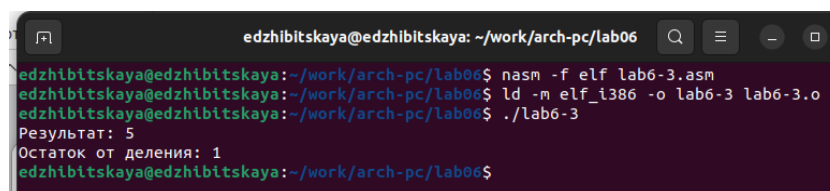
```
edzhibitskaya@edzhibitskaya:~/work/arch-pc/lab06$ gedit lab6-3.asm
edzhibitskaya@edzhibitskaya:~/work/arch-pc/lab06$

Открыть ▾ [F1] *lab6-3.a ~/work/arch-p

1 %include 'in_out.asm' ; подключение внешнего файла
2 SECTION .data
3 div: DB 'Результат: ',0
4 rem: DB 'Остаток от деления: ',0
5 SECTION .text
6 GLOBAL _start
7 _start:
8 ; ---- Вычисление выражения
9 mov eax,4 ; EAX=4
10 mov ebx,6 ; EBX=6
11 mul ebx ; EAX=EAX*EBX
12 add eax,2 ; EAX=EAX+2
13 xor edx,edx ; обнуляем EDX для корректной работы div
14 mov ebx,5 ; EBX=5
15 div ebx ; EAX=EAX/5, EDX=остаток от деления
16 mov edi,eax ; запись результата вычисления в 'edi'
17 ; ---- Вывод результата на экран
18 mov eax,div ; вызов подпрограммы печати
19 call sprint ; сообщения 'Результат: '
20 mov eax,edi ; вызов подпрограммы печати значения
21 call iprintLF ; из 'edi' в виде символов
22 mov eax,rem ; вызов подпрограммы печати
23 call sprint ; сообщения 'Остаток от деления: '
24 mov eax,edx ; вызов подпрограммы печати значения
25 call iprintLF ; из 'edx' (остаток) в виде символов
26 call quit ; вызов подпрограммы завершения
```

Рис. 2.9: Изменение текста программы lab6-3

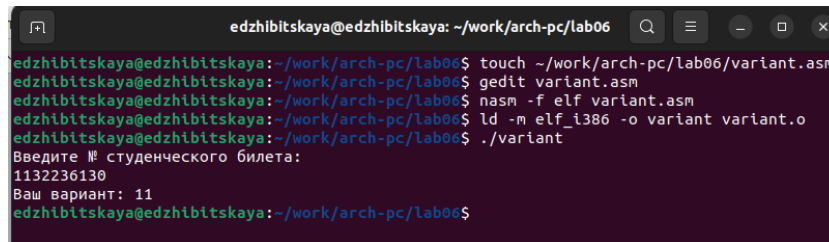
Затем запустим и проверим корректность работы (рис. 2.10).



```
edzhibitskaya@edzhibitskaya: ~/work/arch-pc/lab06
edzhibitskaya@edzhibitskaya:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
edzhibitskaya@edzhibitskaya:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
edzhibitskaya@edzhibitskaya:~/work/arch-pc/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1
edzhibitskaya@edzhibitskaya:~/work/arch-pc/lab06$
```

Рис. 2.10: Запуск измененной программы lab6-3

В конце создадим файл variant.asm, заполним его текстом из Листинг 6.4, создадим исполняемый файл и запустим (рис. 2.11).



```
edzhbitskaya@edzhbitskaya: ~/work/arch-pc/lab06
edzhbitskaya@edzhbitskaya: ~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/variant.asm
edzhbitskaya@edzhbitskaya: ~/work/arch-pc/lab06$ gedit variant.asm
edzhbitskaya@edzhbitskaya: ~/work/arch-pc/lab06$ nasm -f elf variant.asm
edzhbitskaya@edzhbitskaya: ~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
edzhbitskaya@edzhbitskaya: ~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132236130
Ваш вариант: 11
edzhbitskaya@edzhbitskaya: ~/work/arch-pc/lab06$
```

Рис. 2.11: Запуск variant

Далее ответим на следующие вопросы:

1. Какие строки листинга 6.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’?

```
mov eax,rem call sprint
```

2. Для чего используются следующие инструкции? `mov ecx, x` `mov edx, 80` `call sread`

Для считывания переменной `x`.

3. Для чего используется инструкция “`call atoi`”?

Для преобразования `ascii`-код символа в целое число.

4. Какие строки листинга 6.4 отвечают за вычисления варианта?

```
xor edx,edx mov ebx,20 div ebx inc edx
```

5. В какой регистр записывается остаток от деления при выполнении инструкции “`div ebx`”?

АН

6. Для чего используется инструкция “`inc edx`”?

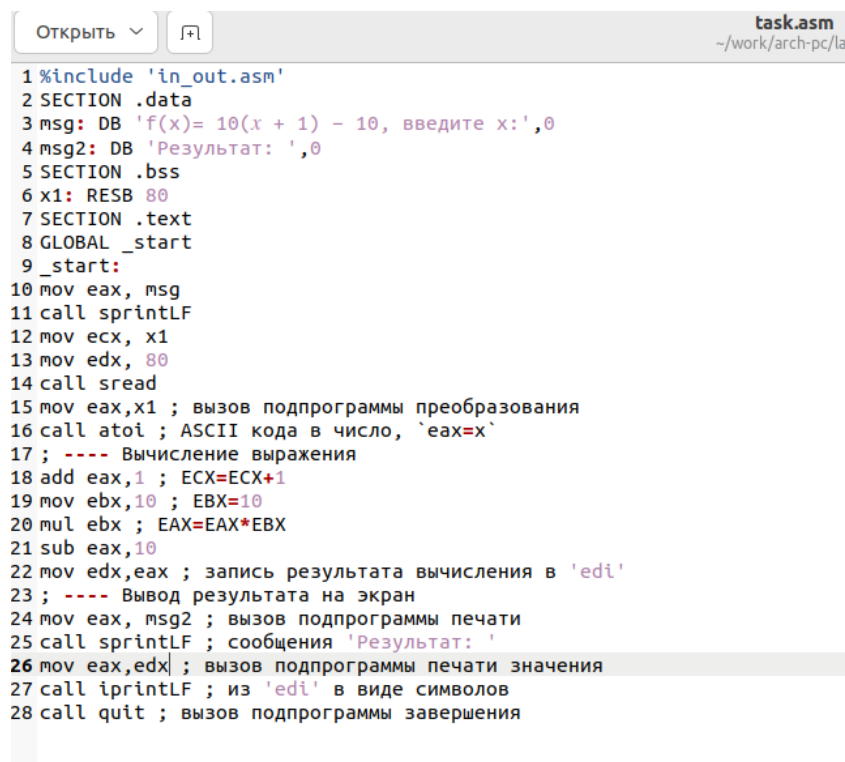
Для прибавления единицы к значению регистра `ebx`.

7. Какие строки листинга 6.4 отвечают за вывод на экран результата вычислений?

```
mov eax,edx call iprintLF
```

3 Задание для самостоятельной работы

Создадим файл task.asm для выполнения задания, напишем туда программу для вычисления выражения $10(x + 1) - 10$, $x = 1$, $x = 7$ (рис. 3.1).



```
task.asm
~/work/arch-pc/la

1 %include 'in_out.asm'
2 SECTION .data
3 msg: DB 'f(x)= 10(x + 1) - 10, введите x:',0
4 msg2: DB 'Результат: ',0
5 SECTION .bss
6 x1: RESB 80
7 SECTION .text
8 GLOBAL _start
9 _start:
10 mov eax, msg
11 call sprintf
12 mov ecx, x1
13 mov edx, 80
14 call sread
15 mov eax,x1 ; вызов подпрограммы преобразования
16 call atoi ; ASCII кода в число, `eax=x`
17 ; ---- Вычисление выражения
18 add eax,1 ; ECX=ECX+1
19 mov ebx,10 ; EBX=10
20 mul ebx ; EAX=EAX*EBX
21 sub eax,10
22 mov edx,eax ; запись результата вычисления в `edi`
23 ; ---- Вывод результата на экран
24 mov eax, msg2 ; вызов подпрограммы печати
25 call sprintf ; сообщения 'Результат: '
26 mov eax,edx ; вызов подпрограммы печати значения
27 call iprintf ; из `edi` в виде символов
28 call quit ; вызов подпрограммы завершения
```

Рис. 3.1: Написание программы

Запустим файл и убедимся в корректности работы (рис. 3.2).

```
edzhibitskaya@edzhibitskaya: ~/work/arch-pc/lab06
edzhibitskaya@edzhibitskaya:~/work/arch-pc/lab06$ nasm -f elf task.asm
edzhibitskaya@edzhibitskaya:~/work/arch-pc/lab06$ ld -m elf_i386 -o task task.o
edzhibitskaya@edzhibitskaya:~/work/arch-pc/lab06$ ./task
f(x)= 10(x + 1) - 10, введите x:
1
Результат:
10
edzhibitskaya@edzhibitskaya:~/work/arch-pc/lab06$ nasm -f elf task.asm
edzhibitskaya@edzhibitskaya:~/work/arch-pc/lab06$ ld -m elf_i386 -o task task.o
edzhibitskaya@edzhibitskaya:~/work/arch-pc/lab06$ ./task
f(x)= 10(x + 1) - 10, введите x:
7
Результат:
70
edzhibitskaya@edzhibitskaya:~/work/arch-pc/lab06$
```

Рис. 3.2: Запуск кода

4 Выводы

В ходе работы было произведено знакомство с арифметическими инструкциями языка ассемблера NASM, изучены некоторые его особенности, а также написана небольшая программа вычисления выражения.