

Лабораторная работа №4

Команды передачи управления

Цель работы: изучение команд перехода для организации циклов и ветвлений, получение навыков создания процедур с аргументами.

Теоретические сведения

JMP <метка>, JMP <адрес>. Команда безусловного перехода (аналог **goto** в языках высокого уровня). В следующем примере после команды **JMP j1** выполняется команда **MOV ECX, 5**:

```
MOV AX, CX                JMP EBX ; Переход по адресу,  
JMP j1                    записанному в EBX  
XOR ECX, ECX  
j1:  MOV ECX, 5
```

В системе команд процессора есть набор команд условного перехода, которые передают управление на выполнение команды по заданному адресу или метки в зависимости от установленных флагов. В противном случае выполняется следующая за командой перехода команда. К примеру, если **EDI=0**, то происходит переход на команду **MOV AX, 4**:

```
DEC EDI  
JZ j1  
XOR EDI, EDI  
j1:  MOV AX, 4
```

Очень часто команды безусловного перехода применяются сразу после команды сравнения **CMR <операнд_1>, <операнд_2>**. Данная команда выполняет вычитание также, как и команда **SUB**, но она только устанавливает флаги и не изменяет значения операндов. В следующей таблице представлен список команд условного перехода, используемых после команды **CMR**, и соответствующие условия перехода.

Тип операндов	Мнемокод	Критерий условного перехода	Значения флагов для перехода
Любые	JE	операнд_1 = операнд_2	ZF = 1
Любые	JNE	операнд_1 ≠ операнд_2	ZF = 0
Знаковые	JL/JNGE	операнд_1 < операнд_2	SF ≠ OF
Знаковые	JLE/JNG	операнд_1 ≤ операнд_2	SF ≠ OF или ZF = 1
Знаковые	JG/JNLE	операнд_1 > операнд_2	SF = OF и ZF = 0
Знаковые	JGE/JNL	операнд_1 ≥ операнд_2	SF = OF
Беззнаковые	JB/JNAE	операнд_1 < операнд_2	CF = 1
Беззнаковые	JBE/JNA	операнд_1 ≤ операнд_2	CF = 1 или ZF = 1
Беззнаковые	JA/JNBE	операнд_1 > операнд_2	CF = 0 и ZF = 0
Беззнаковые	JAE/JNB	операнд_1 ≥ операнд_2	CF = 0

В общем случае, действие команд условного перехода зависит от состояния флагов, которое устанавливает предшествующая команда. Следующие команды действуют в зависимости от состояния конкретного флага:

Название флага	Команда условного перехода	Состояние флага
Переноса CF	JC	CF=1
Чётности PF	JP	PF=1
Нуля ZF	JZ	ZF=1
Знака SF	JS	SF=1
Переполнения OF	JO	OF=1
Переноса CF	JNC	CF=0
Чётности PF	JNP	PF=0
Нуля ZF	JNZ	ZF=0
Знака SF	JNS	SF=0
Переполнения OF	JNO	OF=0

Команды **JCXZ/JECXZ** передают управление по метке в случае, если значение регистра **CX/ECX=0**. Данные команды удобно использовать для организации циклов, если регистр **ECX** используется как счётчик. Команды перехода изменяют последовательность выполнения команд за счёт изменения содержимого регистра **EIP**.

Подобно циклам *for* на языках высокого уровня на ассемблере можно создавать циклы с помощью следующих команд:

LOOP <метка>

Данная команда сначала выполняет уменьшение на единицу содержимого регистра **CX/ECX**. Затем она производит сравнение: если **CX/ECX>0**, то управление передаётся на метку. Если **CX/ECX=0**, то управление передаётся следующей за **LOOP** команде.

Пример:	
Сложение квадратов чисел от 1 до 10:	
$10^2 + 9^2 + \dots + 1^2$	<pre> MOV ECX, 10 XOR EBX, EBX j1: MOV EAX, ECX IMUL EAX ADD EBX, EAX LOOP j1 </pre>

LOOPE/LOOPZ <метка>

Команда уменьшает значение **CX/ECX** на единицу. Если **CX/ECX>0** и **ZF=1**, то управление передаётся на метку перехода. Если **CX/ECX=0** или **ZF=0**, то выполняется следующая за **LOOP** команда.

LOOPNE/LOOPNZ <метка>

Действие команды аналогично предыдущей, но выход из цикла происходит, если **CX/ECX=0** или **ZF=1**. Последние две команды удобно использовать, если необходимо организовать досрочный выход из цикла. Недостатком данных команд **LOOPcc** является невозможность делать большие переходы, т.е. смещение метки относительно текущего значения **EIP** не может быть больше 128 байт.

Пример:	
Поиск элемента <i>a</i> в массиве <i>mas</i> с конца. Если элемент найден, происходит выход из цикла	<pre> mas dw 1, 2, 3, 5, 7, 8, 9, 10, -8, 5 a dw 7 ... MOV ECX, 10 MOV AX, a j1: CMP mas[ECX*2-2], AX LOOPNE j1 </pre>

Команды для работы со стеком

PUSH <операнд>. Команда сначала уменьшает значение **ESP** на 2/4, в зависимости от размера помещаемых данных, а затем записывает в вершину стека по адресу **ESP** значение операнда размером 2/4 байта.

POP <операнд>. Команда сначала выталкивает 2/4 байта, начинающиеся с адреса **ESP**, из стека в регистр или ячейку памяти, а затем увеличивает значение регистра **ESP** на 2/4, в зависимости от размера пересылаемых данных.

PUSHA/PUSHAD. Данная команда не имеет операндов. Она сначала уменьшает значение **SP/ESP** на 16/32, а затем помещает в стек содержимое регистров общего назначения в следующем порядке: **AX/EAX**, **CX/ECX**, **DX/EDX**, **BX/EBX**, **SP/ESP**, **BP/EBP**, **SI/ESI**, **DI/EDI**.

POPA/POPAD. Команда выталкивает из стека значения регистров общего назначения в следующем порядке: **DI/EDI**, **SI/ESI**, **BP/EBP**, **SP/ESP**, **BX/EBX**, **DX/EDX**, **CX/ECX**, **AX/EAX**. Содержимое **SP/ESP** при этом не восстанавливается. Затем значение **ESP** увеличивается на 16/32.

CALL <имя_подпрограммы>. Команда предназначена для вызова подпрограммы. Она помещает адрес следующей после **CALL** команды в стек и передаёт управление по адресу подпрограммы.

RET <операнд>. Осуществляет возврат в вызывающую подпрограмму по адресу, хранящему на вершине стека. Команда извлекает из стека адрес возврата, помещает его в **EIP**, и увеличивает содержимое **ESP** на значение, которое имеет непосредственный операнд, выталкивая тем самым из стека данные.

Существуют определённые правила организации подпрограмм и передачи им аргументов. В операционной системе Windows принято вызывать подпрограммы, предварительно передавая их аргументы в стек.

Пример вызова функции с тремя аргументами	<pre>PUSH EAX PUSH DX PUSH 100 CALL f</pre>
---	---

Допустим изначально **ESP=0002FFFC**. Тогда после выполнения данного фрагмента кода после входа в подпрограмму, стек будет содержать следующую информацию:

Адреса ячеек ОП		Содержимое
ESP=0002FFEE →	0002FFEE-0002FFF1	Адрес возврата (EIP)
	0002FFF2-0002FFF5	100
	0002FFF6-0002FFF7	DX
	0002FFF8-0002FFFB	EAX
	0002FFFC-0002FFFF	

После выполнения подпрограммы стек должен быть возвращён в то состояние, которое он имел перед выполнением первой команды **PUSH EAX**. Очистить стек от аргументов обязана вызываемая подпрограмма путём выполнения команды **RET 10**. 10 байт – суммарный размер аргументов. Если подпрограмма использует для вычислений какие-либо регистры, то необходимо сохранить их содержимое в начале подпрограммы в стеке, а в конце подпрограммы восстановить из стека. Таким образом, для вызывающей подпрограммы значения регистров не изменятся. Для удобства можно задать регистру-указателю кадры стека **EBP** такое значение, чтобы он указывал на последний помещённый в стек аргумент.

Если подпрограмма является функцией и возвращает какие-либо значения, то их необходимо поместить в регистр **AL/AX/EAX/EDX:EAX**, если возвращаемое значение имеет размер соответственно 1/2/4/8 байт. Если функция возвращает, к примеру, число типа **long** либо **double** размером 8 байт, то старшую половину результата нужно поместить в **EDX**, младшую – в **EAX**. Остальные регистры, которые не используются для возврата значений из подпрограммы, требуется восстанавливать до исходного состояния, которое они имели в начале подпрограммы.

<p>Пример функции с тремя аргументами <code>int f(int x, int_16 y, int z),</code> вычисляющей значение выражения $xy + z$. x, z – знаковые 4-байтовые аргументы, y – знаковый 2-байтовый. Аргументы при этом передаются в стек в обратном порядке:</p> <p><code>PUSH EAX ; z (4 байта)</code> <code>PUSH DX ; y (2 байта)</code> <code>PUSH 100 ; x (4 байта)</code> <code>CALL f</code></p>	<pre>f proc PUSH EBX ; EBX сохранить в стеке PUSH EBP ; EBP сохранить в стеке MOV EBP, ESP ; В EBP скопировать ESP ADD EBP, 12 ; Сместить EBP ближе к аргументам MOV AX, [EBP+4] ; В AX поместить второй аргумент y CWDE MOV EBX, [EBP] ; В EBX поместить первый аргумент x IMUL EBX ADD EAX, [EBP+6] ; К EAX прибавить третий аргумент z POP EBP ; Восстановить EBP POP EBX ; Восстановить EBX RET 10 ; Очистка стека от аргументов и возврат в вызывающую подпрограмму f endp ; Результат – в EAX</pre>
---	--

Задания для выполнения к работе

1. Написать программу для вычисления значения арифметического выражения, используя команды условного и безусловного перехода согласно варианту задания. Подобрать набор тестовых данных (не менее 3). При выполнении операций с числами, преобразовывать их к 4-байтовым числам со знаком. Результат вывести на экран.
2. Написать программу для вычисления значения арифметического выражения, содержащего функцию. Вычисление функции организовать в виде отдельной подпрограммы по всем правилам, описанным выше. Для обработки массивов использовать команды для работы с циклами и команды условного перехода. Подобрать набор тестовых данных (не менее 3). Результат вывести на экран.

Пример выполнения первого задания:

#	$a = \begin{cases} x + y, & x \leq 5 \\ 5xy, & x \in (5,10) \\ z^2 + y, & x \geq 10 \end{cases}$	x – знаковое однобайтовое, y – беззнаковое двухбайтовое, z – знаковое 4-байтовое
---	--	--

Программа на ассемблере, вычисляющая данное выражение, имеет вид:

```
.386
.model flat, stdcall
option casemap: none

include d:\masm32\include\kernel32.inc
include d:\masm32\include\msvcrt.inc

includelib d:\masm32\lib\kernel32.lib
includelib d:\masm32\lib\msvcrt.lib

.data
    x db -10
    y dw 40
    z dd -100
    format db "a = %d", 0
.code
start:
    XOR ECX, ECX    ; Обнулить ECX
    MOV CX, y        ; Поместить y в младшую половину ECX. y – беззнаковое,
                    ; поэтому расширять его не нужно. Достаточно обнулить старшие байты.
    ; В ECX – 4-байтовое
    MOV AL, x        ; В AL поместить x
    CMP AL, 10       ; Если AL >= 10, то переход на j2
    JGE j2
    CBW              ; Расширение x до 2 байт
    CWDE             ; Расширение x до 4 байт
    CMP AL, 5
    JLE j1           ; Если AL <= 5, то переход на j1
    IMUL ECX         ; EDX:EAX = x*y
    IMUL EAX, 5      ; EAX = 5*x*y
    JMP j_out
j1:
    ADD EAX, ECX     ; EAX = x + y
    JMP j_out
j2:
    MOV EAX, z       ; EAX = z
    IMUL z           ; EDX:EAX = z^2
    ADD EAX, ECX     ; EAX = z^2 + y
j_out:
    push EAX
    push offset format
    call crt_printf    ; Вывод результата на экран

    call crt__getch    ; Задержка ввода
    push 0
    call ExitProcess   ; Выход из программы
end start
```

Набор тестовых данных

x	y	z	a
-10	40	10	30
7	50	100	30
100	100	-100	10100

Пример выполнения второго задания:

#	$a = \sum_{i=0}^n x_i^2 + \frac{f(y_i)}{h},$ $f(y) = y^2 + 50$	<p>x – массив 2-байтовых знаковых чисел</p> <p>y – массив однобайтовых знаковых чисел</p> <p>h – двубайтовое знаковое</p> <p>n – беззнаковая переменная размером 2 байта</p>
---	--	--

Ниже приведён текст программы для вычисления данного выражения:

```
.386
.model flat, stdcall
option casemap: none

include d:\masm32\include\kernel32.inc
include d:\masm32\include\msvcrt.inc

includelib d:\masm32\lib\kernel32.lib
includelib d:\masm32\lib\msvcrt.lib

.data
    x dw -10, -2000, 12, 15, 5, -50, 170, 8, 45, 10
    y db 100, -100, 7, -15, 30, 20, 35, 40, 10, 10
    h dw -50
    n dw 3
    format db "a = %d", 0

.code
; Функция от одного аргумента
f proc
    MOV AL, [ESP+4] ; Аргумент смещён относительно вершины стека на 4 байта
    CBW             ; Расширение аргумента до 2 байт
    CWDE            ; До 4 байт
    IMUL EAX         ; Вычисление квадрата EAX = y²
    ADD EAX, 50      ; EAX = y² + 50
    ret 2 ; Выталкивание из стека 2 байт и возврат в основную программу
f endp

start:
    XOR EBX, EBX ; В EBX будет накапливаться сумма
    XOR ESI, ESI ; ESI - индекс i элементов в массивах
    XOR ECX, ECX ; ECX - счётчик итераций
    MOV CX, n    ; CX = n
j1:
    MOV AX, x[ESI*2] ; AX = x_i
    CWDE            ; Т.к. x - беззнаковое, нужно расширить его до 4 байт
    IMUL EAX        ; EAX = x_i²
    ADD EBX, EAX    ; EBX = EBX + x_i²

    MOV AX, h
    CWDE            ; Расширение h до размерности двойного слова
    MOV EDI, EAX    ; Переместить делитель h в EDI

    PUSH word ptr y[ESI] ; Поместить в стек y_i. Поскольку команда PUSH не
    ; может поместить в стек 1 байт, то помещаем 2 байта, т.е. y_i и y_{i+1}. Но в
    ; процедуре используем только y_i
    call f          ; Вызов функции. Результат: EAX = x_i² + 50
    CDQ             ; Расширение делимого до 8 байт перед операцией деления
    IDIV EDI        ; EAX = f(y_i) / h. В EDX - остаток от деления
    ADD EBX, EAX    ; EBX = EBX + f(y_i) / h
    INC ESI         ; i = i + 1
```

```

LOOP j1      ; ECX = ECX - 1. Переход в начало цикла, если ECX ≠ 0
MOV EAX, EBX ; Поместить результат в EAX

push EAX
push offset format
call crt_printf ; Вывод результата на экран

call crt__getch ; Задержка ввода
push 0
call ExitProcess ; Выход из программы
end start

```

Набор тестовых данных

x = {-10}	y = {100}	h = -50	n = 1	a = -101
x = {-10, 3}	y = {10, 3}	h = 100	n = 2	a = 110
x = {20, 1, 1000}	y = {10, 5, 100}	h = -10	n = 3	a = 1001428

Варианты заданий задачи №1

Вариант	Выражение	Размерность и тип переменных
1	$a = \begin{cases} xy + 4, & x < 10 \text{ и } y < 2 \\ \frac{x}{z} - 6, & x < 10 \text{ и } y \geq 2 \\ (xy)^2, & x \geq 10 \end{cases}$	x, y, z – знаковые числа размером 2 байта
2	$b = \begin{cases} \frac{x+y}{z-y}, & x < 10 \text{ и } y < 2 \\ xy - 6, & x < 10 \text{ и } y \geq 2 \\ (x+y)^2, & x \geq 10 \end{cases}$	x – беззнаковое число размером один байт y, z – знаковые числа размером 2 байта
3	$e = \begin{cases} \frac{x+y}{z-y}, & x > 10 \\ xy - 6, & x \leq 10 \text{ и } z > 2 \\ (x+z)^2, & x \leq 10 \text{ и } z \leq 2 \end{cases}$	x – знаковое 4-байтовое y – беззнаковое 2-байтовое z – знаковое однобайтовое
4	$q = \begin{cases} x^2 + y^2, & x \in [5, 15] \\ x^2 - z^2, & x \notin [5, 15] \text{ и } z > 0 \\ 500 \frac{x^2 + y^2}{x^2 - z^2}, & x \notin [5, 15] \text{ и } z \leq 0 \end{cases}$	x – беззнаковое однобайтовое y, z – знаковое 2-байтовое
5	$a = \begin{cases} \frac{y}{z} + 100, & z \leq 10 \text{ или } x \in [0; 3] \\ xyz - 1, & z > 10 \text{ и } x \notin [0; 3] \text{ и } y > 0 \\ z^3, & z > 10 \text{ и } x \notin [0; 3] \text{ и } y \leq 0 \end{cases}$	x – беззнаковое 2-байтовое y – знаковое 4-байтовое z – знаковое однобайтовое
6	$b = \begin{cases} \frac{x}{y} + 10 \frac{x}{z}, & z \leq 0 \text{ и } 1 \leq x < 3, \\ -x + y^2, & z > 0, \\ 23 - xyz, & z \leq 0 \text{ и } x \notin [1, 3) \end{cases}$	x – знаковое однобайтовое y, z – беззнаковое однобайтовое
7	$t = \begin{cases} \frac{x}{y} + 10 \frac{x}{z}, & x < 1 \text{ и } t < 0, \\ \frac{x+y}{t+3}, & t > 0, \\ \frac{x^2 + 23}{t} - y, & x \geq 1 \text{ и } t < 0 \end{cases}$	x – знаковое однобайтовое y, t – беззнаковое 2-байтовое
8	$a = \begin{cases} x^2 + 32 - xz, & x + y + z > 0, \\ z + y, & x + y + z \leq 0 \text{ и } x > 0, \\ x + y \frac{z}{2}, & x + y + z \leq 0 \text{ и } x \leq 0 \end{cases}$	x – знаковое однобайтовое y – знаковое 4-байтовое z – беззнаковое 2-байтовое
9	$d = \begin{cases} x^2 y^3 z^4, & z \notin [-1; 1] \text{ и } y < 0 \\ z + 1 + xy, & y \geq 0 \\ 4x - 2z, & z \in [-1; 1] \text{ и } y < 0 \end{cases}$	x – знаковое 2-байтовое y, z – беззнаковое однобайтовое
10	$r = \begin{cases} 12z - 3xy, & x > 5 \text{ и } y > 8, \\ \frac{y}{2} + \frac{z}{5}, & x > 5 \text{ и } y \leq 8, \\ \frac{(z+1)^2}{y}, & x \leq 5 \end{cases}$	x – знаковое 2-байтовое y – беззнаковое однобайтовое z – беззнаковое 2-байтовое
11	$s = \begin{cases} \frac{42x}{3} y^3, & x < -15 \text{ и } y \in [-10; 20], \\ z + \frac{1}{y} + x, & x < -15 \text{ и } y \notin [-10; 20], \\ 3x + z^3, & x \geq -15 \end{cases}$	x – знаковое 4-байтовое y – знаковое 2-байтовое z – беззнаковое однобайтовое

12	$d = \begin{cases} \frac{1}{x+y^2} + z^3, & x \geq 0, \\ z + 1 + xy, & z \in [2,8] \text{ и } x < 0, \\ 7x - 2z, & z \notin [2,8] \text{ и } x < 0 \end{cases}$	x, z – знаковое однобайтовое y – знаковое 4-байтовое
13	$e = \begin{cases} \frac{(x+y+1)^2}{y^3-2} z^4, & y > 10 \text{ и } z > 5, \\ -z^2 + \frac{1}{x} + \frac{x}{y+10}, & z \in [-5;5] \text{ и } y > 10, \\ -5x^2 - 2(z+1)y, & y \leq 10 \end{cases}$	x – беззнаковое 2-байтовое y – знаковое однобайтовое z – беззнаковое однобайтовое
14	$p = \begin{cases} x^2 + 1 + z, & x > 3, \\ \frac{x}{(4-z^2)}, & x \in [-3;3] \text{ и } z \notin [-2;2], \\ x - y - 2z, & x \in [-3;3] \text{ и } z \in [-2;2] \end{cases}$	x – знаковое однобайтовое y – знаковое 2-байтовое z – беззнаковое 2-байтовое
15	$a = \begin{cases} \frac{(x+2)^2}{(y-2)^3}, & y \in [4;50], \\ \frac{x+y+z}{y}, & (y > 50 \text{ или } y < 4) \text{ и } z > 10, \\ 19x - 21y, & (y > 50 \text{ или } y < 4) \text{ и } z \leq 10 \end{cases}$	x, z – знаковое 4-байтовое y – знаковое 2-байтовое
16	$d = \begin{cases} \frac{x^4 - z}{y - x} + 12 \frac{x}{z + x}, & x \in [0,15], \\ \frac{x}{t + z^2}, & x < 0, \\ \frac{x^2 + 23}{t} - y, & x > 15 \end{cases}$	x – беззнаковое однобайтовое y, z, t – знаковые 2-байтовые числа.
17	$a = \begin{cases} \frac{x^2 + y^2}{x^3 - z^2} + x, & x \in [-1,9], \\ x^2, x \notin [-1,9] \text{ и } z > 0 \text{ и } k < 10, \\ 7 \frac{y^4}{x^2}, x \notin [-1,9] \text{ и } (z \leq 0 \text{ или } k \geq 10) \end{cases}$	x, y, z – знаковые числа размером 2 байта k – беззнаковое однобайтовое число
18	$p = \begin{cases} \frac{4-x}{z} + y^4, & x < 5 \text{ и } y < 10, \\ \frac{x^3 - z^2 - y}{x - 23} y^4, & y \geq 10, \\ (xy)^2 z, & x \geq 5 \text{ и } y < 10 \end{cases}$	x, y, z – знаковые числа размером 2 байта
19	$h = \begin{cases} 7 - x \frac{4 - (xz)^2}{z} + 23, & k \in (2,10), \\ xyz - 1, & k \geq 10 \text{ и } k \leq 2 \text{ и } z > 10, \\ x^3 - z^3, & k \geq 10 \text{ и } k \leq 2 \text{ и } z \leq 10 \end{cases}$	x, y, z – знаковые числа размером один байт k – беззнаковое 2-байтовое число
20	$b = \begin{cases} \frac{y + x^3}{xyz - 1 - y}, & x + y \in [5,15], \\ z^3 y - 15, & x + y > 15, \\ \frac{x - 23}{z^4}, & x + y < 5 \end{cases}$	x, y, z – знаковые числа размером 2 байта

21	$a = \begin{cases} \left(\frac{x}{z}\right)^2 + 100yz , & x + z < y \\ 500 \frac{x^4}{y}, & x + z \geq y \text{ и } x > 0 \\ x^2 - y^2 + 4, & x + z \geq y \text{ и } x \leq 0 \end{cases}$	x – знаковое 2-байтовое y – знаковое 4-байтовое z – беззнаковое однобайтовое
22	$b = \begin{cases} \frac{x^2 - z^2}{y}, & x > 10 \text{ и } y > 0, \\ -xyz, & x > 10 \text{ и } y \leq 0 \text{ и } z > 0, \\ \frac{ x + y }{35}, & x > 10 \text{ и } y \leq 0 \text{ и } z \leq 0 \text{ или } x \leq 10 \end{cases}$	x – знаковое 2-байтовое y – знаковое 4-байтовое z – беззнаковое 2-байтовое
23	$t = \begin{cases} x^2 + 3y^3, & y < 5 \text{ или } y > 40, \\ \frac{z - 4x}{5 - 6y}, & y \in [5, 20], \\ xy + 6xz , & y \in (20, 40] \end{cases}$	x – беззнаковое 2-байтовое y, z – знаковое 4-байтовое
24	$r = \begin{cases} \left(\frac{x}{y - z}\right)^2, & x > 5 \text{ и } z > 15, \\ \frac{x + yz}{45}, & x \leq 5 \text{ или } z \leq 10, \\ xy^2 + zx^2 - xyz , & x \leq 5 \text{ или } z \in (10, 15] \end{cases}$	x – беззнаковое однобайтовое y – знаковое 2-байтовое z – беззнаковое 2-байтовое
25	$s = \begin{cases} \left(\frac{x - y^2}{5}\right)^3, & x - y > 0, \\ \frac{x + y}{5y + z}, & x - y \leq 0 \text{ и } z > 10, \\ x^2 + y^2 + z^2, & x - y \leq 0 \text{ и } z \leq 10 \end{cases}$	x – знаковое 2-байтовое y – беззнаковое однобайтовое z – знаковое однобайтовое

Варианты заданий задачи №2

1	$a = \sum_{i=0}^n i^2 + \frac{x_i^2}{y_i^2} - y_i^k + f^3(x_i, y_i),$ $f(x, y) = \begin{cases} x + y , & x > 0, \\ x - y , & x \leq 0 \end{cases}$	x – массив 2-байтовых беззнаковых чисел y – массив 4-байтовых знаковых чисел $k \in [0, 10]$ n – беззнаковая переменная размером 4 байта
2	$s = \sum_{i=0}^n 100 \frac{i}{h} + \frac{y_i}{x_i} + g(x_i, y_i),$ $g(x, y) = \begin{cases} x^2 - y^x, & x \in (1; 7), \\ x^2 + y^2 + xy , & x \notin (1; 7) \end{cases}$	h – знаковая переменная размером один байт x – массив 1-байтовых беззнаковых чисел y – массив 2-байтовых беззнаковых чисел n – беззнаковая переменная размером 2 байта

3	$a = \sum_{i=0}^n ih + 7 - x_i \frac{y_i + h}{i} + g(x_i, y_i),$ $g(x, y) = \begin{cases} y^x, & x > y \text{ и } x \in (1; 5) \\ \frac{x^2}{4} + x y, & x \leq y \text{ или } x \notin (1; 5) \end{cases}$	<p>h – беззнаковая переменная размером 2 байта</p> <p>x – массив беззнаковых 1-байтовых чисел</p> <p>y – массив знаковых 2-байтовых чисел</p> <p>n – беззнаковая переменная размером 4 байта</p>
4	$a = \sum_{i=0}^n x_i + 200 \frac{i^4 h}{y_i x_i} + \frac{p(x_i, y_i)}{50},$ $p(x, y) = \begin{cases} 5000 x + 2^{(x+y)}, & (x + y) \in [0, 20] \\ 2500 y - y^2, & (x + y) \notin [0, 20] \end{cases}$	<p>h – знаковая переменная размером 4 байта</p> <p>x – массив знаковых 4-байтовых чисел</p> <p>y – массив беззнаковых 2-байтовых чисел</p> <p>n – беззнаковая переменная размером 1 байт</p>
5	$a = \sum_{i=0}^n \begin{cases} hx_i - \frac{y_i}{4} + f(x_i + 5, y_i), & i - \text{чётно}, \\ \frac{x_i}{y_i} - h \frac{y_i x_i}{x_i + y_i} + g(y_i), & i - \text{нечётно}, \end{cases}$ $f(x, y) = \begin{cases} \frac{x^y}{ y } & x \in [2, 4] \text{ и } y \in [0, 10], \\ x + y & x \notin [2, 4] \text{ или } y \notin [0, 10], \end{cases}$ $g(y) = 5y^2 + 10y + 1$	<p>h – беззнаковая переменная размером 2 байта</p> <p>x – массив 4-байтовых знаковых чисел</p> <p>y – массив 2-байтовых беззнаковых чисел</p> <p>n – беззнаковая переменная размером 2 байта</p>
6	$a = \sum_{i=0}^n \begin{cases} h \frac{i}{y_i} + x_i + g(x_i, y_i), & i - \text{нечетно}, \\ \frac{x_i^3}{y_i} h - \left(\frac{y_i x_i}{x_i - y_i} \right)^3, & i - \text{четно}, \end{cases}$ $g(x, y) = \frac{ x + y + 50 }{5}$	<p>h – знаковая переменная размером 2 байта</p> <p>x – массив беззнаковых 1-байтовых чисел</p> <p>y – массив знаковых 2-байтовых чисел</p> <p>n – беззнаковая переменная размером 4 байта</p>
7	$a = \sum_{i=0}^{i=m} \sum_{j=1}^n \frac{3}{x_i} - \frac{y_j}{x_i - h} + t(x_i, y_j),$ $t(x, y) = \begin{cases} x^2 + 5, & y > (x + 1), \\ x^2 - 1 , & y \leq (x + 1) \end{cases}$	<p>h – знаковая переменная размером 2 байта</p> <p>x – массив беззнаковых 1-байтовых чисел</p> <p>y – массив 2-байтовых знаковых чисел</p> <p>m, n – беззнаковые переменные размером 1 байт</p>
8	$a = \sum_{i=0}^m \sum_{j=0}^n \frac{x_i}{k(i, j)} - \frac{y_i^3 x_i i + 6}{x_i^2},$ $k(i, j) = \begin{cases} \frac{i}{3}, & i \text{ кратно } 3, \\ 5 \frac{i + j}{ i - j }, & i \text{ не кратно } 3 \end{cases}$	<p>h – беззнаковая переменная размером 2 байта</p> <p>x – массив знаковых 2-байтовых чисел</p> <p>y – массив знаковых 4-байтовых чисел</p> <p>m, n – беззнаковые переменные размером 2 байта</p>
9	$a = \sum_{i=0}^m \sum_{j=0}^m x_i^5 - ih + y_i(h + 17i) f(x_i, y_j),$ $f(x, y) = \begin{cases} x + y, & x > 10, \\ \frac{1000}{x - y}, & x \leq 10 \end{cases}$	<p>h – знаковая переменная размером 1 байт</p> <p>x – массив 1-байтовых знаковых чисел</p> <p>y – массив беззнаковых 2-байтовых чисел</p> <p>m – беззнаковая переменная размером 2 байта</p>

10	$b = \sum_{i=0}^n (x_i + y_i)^k + 5k - g(x_i, y_j),$ $g(x, y) = \begin{cases} x(x + y), & x > 3 \text{ и } y > 3, \\ 200 x , & x \leq 3 \text{ или } y \leq 3 \end{cases}$	x – массив 1-байтовых знаковых чисел; y – массив 2-байтовых беззнаковых чисел; $k \in [0,10]$ m – беззнаковая переменная размером 4 байта
11	$r = \sum_{i=0}^n x_i + 2x_i y_i^2 + 8y_i^3 + t(x_i, k),$ $t(x, k) = \begin{cases} x^k, & k > 5, \\ (x + 1)^k, & k \leq 5 \end{cases}$	x, y — массивы 2-байтовых знаковых чисел $k \in [0,10]$ n – беззнаковая переменная размером 1 байт
12	$r = \sum_{i=0}^n \begin{cases} 5x_i^3 + 10 + f_1(x_i, i), & i < \frac{n}{2}, \\ 8y_i^2 + x_i + f_2(y_i, i), & i \geq \frac{n}{2}, \end{cases}$ $f_1(x, k) = \sum_{j=0}^k jx, \quad f_2(y, k) = \sum_{j=0}^k \frac{y}{j}$	x – массив 2-байтовых знаковых чисел y – массив 2-байтовых беззнаковых чисел n – беззнаковая переменная размером 4 байта
13	$r = \sum_{i=0}^n k(y_i + x_i) + k^3 - y_i^2 + \frac{f(x_i, y_i)}{k},$ $f(x, y) = \sum_{j=0}^x yj - 1 $	x – массив 1-байтовых беззнаковых чисел y – массив 2-байтовых знаковых чисел k – беззнаковая переменная размером 2 байта n – беззнаковая переменная размером 1 байт
14	$r = \sum_{i=0}^n x_i + \sum_{i=0}^n \begin{cases} \frac{x_i}{y_i} + f(x_i, y_i), & i - \text{чётно}, \\ x_i + y_i^2 + f(x_i^2, y_i^2), & i - \text{нечётно}, \end{cases}$ $f(x, y) = \begin{cases} \frac{ x + y }{200}, & x > 7, \\ (x - y)^3, & x \leq 7 \end{cases}$	x – массив 4-байтовых знаковых чисел y – массив 2-байтовых беззнаковых чисел n – беззнаковая переменная размером 2 байта
15	$w = \sum_{i=1}^n x_i h + \frac{y_i}{x_i} - \frac{h}{y_i} x_i + f(x_i, y_i),$ $f(x, y) = \begin{cases} (x - 5)^y, & y \in [0,10], \\ (x + y)^2, & y \notin [0,10] \end{cases}$	h – двухбайтовое беззнаковое x – массив однобайтовых беззнаковых чисел y – массив двухбайтовых знаковых чисел n – беззнаковая переменная размером 4 байта
16	$w = \sum_{i=1}^n x^k + \frac{x_i h}{y_i} - \frac{k}{h} (x_i + y_i) + t^2(x_i, y_i),$ $t(x, y) = \begin{cases} x - y^2 , & x > y \text{ и } y > 10, \\ (x + y)^2, & x \leq y \text{ или } y \leq 10 \end{cases}$	h – однобайтовое знаковое x – массив двухбайтовых знаковых чисел y – массив однобайтовых беззнаковых чисел $k \in [0,10]$ n – беззнаковая переменная размером 1 байт
17	$s = \sum_{i=0}^n h \frac{x_i}{y_i} - h y_i + f(x_i, y_i) ,$	h – двухбайтовое беззнаковое x – массив двухбайтовых знаковых чисел y – массив однобайтовых

	$f(x, y) = \begin{cases} xy^2 - 15, & x < y/2, \\ x + \frac{15^4}{ y }, & x \geq \frac{y}{2} \end{cases}$	беззнаковых чисел n – беззнаковая переменная размером 2 байта
18	$v = \sum_{i=1}^n (k + h) \frac{y_i}{hx_i} + (x_i h)^k + \frac{f(x_i)}{f(y_i)},$ $f(x) = \begin{cases} x^2 - 2, & x \in [-5, 5], \\ x^3 + 1, & x \notin [-5, 5] \end{cases}$	h – двухбайтовое знаковое x – массив двухбайтовых беззнаковых чисел y – массив однобайтовых знаковых чисел $k \in [0, 10]$ n – беззнаковая переменная размером 4 байта
19	$s = \sum_{i=1}^m y_i (x_i h)^2 - \left(\frac{h}{y_i} x_i \right) + f(x_i) + f(y_i) ,$ $f(x) = \begin{cases} (x + 1)^3, & x \in (3, 5), \\ \frac{x - 1}{x^2 + 3}, & x \notin (3, 5) \end{cases}$	h – двухбайтовое знаковое x – массив 4-байтовых знаковых чисел y – массив однобайтовых беззнаковых чисел m – беззнаковая переменная размером 1 байт
20	$v = \sum_{i=1}^n h^k + \frac{x_i}{k + y_i} - \frac{x_i^{k+1}}{h} - \frac{y_i + x_i}{hk} + f(x_i, y_i, k),$ $f(x, y, k) = k \frac{ x + y }{x - y}$	h – двухбайтовое знаковое x – массив двухбайтовых беззнаковых чисел y – массив однобайтовых знаковых чисел $k \in [0, 10]$ n – беззнаковая переменная размером 2 байта
21	$d = \sum_{i=0}^n (at_i)^2 + r_i^k - 20 \frac{f(t_i)}{g(r_i)},$ $f(t) = \begin{cases} \frac{5t^2}{t - 2}, & t \text{ кратно } 3, \\ \frac{t - 4}{10}, & t \text{ не кратно } 3, \end{cases} \quad g(r) = r^3 + 2r^2$	t – массив однобайтовых беззнаковых чисел r – массив двухбайтовых знаковых чисел $k \in [0, 7]$ n – беззнаковая переменная размером 4 байта
22	$a = \sum_{i=0}^n 2^k \frac{(x_i + 5)^2}{100} + (y_i^2 - 1)^2 + f(x_i, y_i) ,$ $f(x, y) = \begin{cases} \frac{500}{x}, & x > y \text{ и } x < 10 \\ 500y , & x \leq y \text{ или } x \geq 10 \end{cases}$	x – массив двухбайтовых знаковых чисел y – массив 4-байтовых знаковых чисел $k \in [0, 15]$ n – беззнаковая переменная размером 1 байт
23	$a = \sum_{i=1}^m \sum_{j=1}^n h \left(\frac{x_i + 3y_i}{f(x_i, y_i)} \right)^2 + \left(\frac{x_i - 3y_i}{f(y_i, x_i)} \right)^2,$ $f(x, y) = \begin{cases} x + y , & x \in [-50, 50], \\ \frac{x + 40y}{x}, & x \notin [-50, 50] \end{cases}$	h – двухбайтовое знаковое x – массив 2-байтовых знаковых чисел y – массив однобайтовых знаковых чисел m, n – беззнаковые переменные размером 2 байт
24	$b = \sum_{i=0}^n h \frac{ f(2, t) }{x_i} + 100 \frac{ f(3, t) }{y_i} - x_i y_i,$ $f(r, t) = \begin{cases} r^t, & t \in [0, 10], \\ rt, & t \notin [0, 10] \end{cases}$	h – однобайтовое беззнаковое x – массив однобайтовых знаковых чисел y – массив 2-байтовых беззнаковых чисел n, t – беззнаковые переменные размером 4 байт

25	$a = \sum_{i=0}^m \frac{h^k}{x_i^2} + \frac{f(k, x_i)(h+1)^k}{y_i^2},$ $f(k, x) = \begin{cases} kx + 100, & \frac{x}{10} > k, \\ \frac{x}{k} - 100, & k \geq \frac{x}{10} \end{cases}$	<p>h – 4-байтовое знаковое</p> <p>x – массив 2-байтовых знаковых чисел</p> <p>y – массив однобайтовых беззнаковых чисел</p> <p>$k \in [0, 15]$</p> <p>m – беззнаковая переменная размером 1 байт</p>
----	--	---