Задача 1 Определение значения регистра EAX и флагов

Секция `.data` была размещена в памяти, начиная с базового адреса 0x080ea104. Укажите значение регистра EAX и флагов после выполнения указанных команд секции `.text`.

```
section .data
    a dw 0x2024, 0x2023, 0x2022, 0x2021
    s db "input", 0
    b dd 0xBCEEADDA
    p dd b

section .text
    movzx eax, word[b + 2] ; (1)
    add al, byte[p]        ; (2)
    movzx ax, byte[a + 2]  ; (3)
```

Скопируйте 3 строки ниже в поле ответа и заполните прочерки значениями регистра EAX (в шестнадцатеричном виде) и флагов (0/1) после выполнения соответствующих инструкций:

```
(1) EAX = –
(2) EAX = –, CF = –, OF = –, ZF = –, SF = –
(3) EAX = –
```

Пример возможного формата ответа:

```
(1) EAX = 0xcafebabe
(2) EAX = 0x00000000, CF = 0, OF = 0, ZF = 1, SF = 0
(3) EAX = 0x00cafeba
```

Задача 2-Logic: Логические операции

Для приведённого ниже фрагмента кода на языке Си напишите эквивалентный фрагмент кода на языке ассемблера.

Считайте, что переменные a, b, c, x уже объявлены в секции `.bss`.

Код должен состоять только из инструкций языка ассемблера и меток, реализующих выражение.

В коде **не должно** быть директив `section`, `extern`, метки `main`, инструкций `call` и `ret`.

```
static unsigned int a, b, c, x;
...
x = ++a && --b || c++;
```

Задача 3-Decompile: Декомпиляция программы с массивом

Даны два фрагмента кода на языке Си с шестью пропусками:

```
// объявление глобальных переменных
int* a;
int A[24][пропуск_1][пропуск_2];
int x, y, z;

// выражение-оператор
пропуск_3 = A[пропуск_4][пропуск_5][пропуск_6];
```

Для этих фрагментов компилятор построил следующий код:

```
        mov     ecx, dword [a]
        mov     ebx, dword [y]
        mov     eax, dword [ecx]
        lea     edx, [eax+eax*4]
        mov     eax, ebx
        sal     eax, 4
        sub     eax, ebx
        add     eax, edx
        add     eax, dword [z]
        mov     eax, dword [A+4*eax]
        mov     dword [ecx], eax
```

Вам необходимо восстановить пропуски в исходном фрагменте кода на языке Си. Выпишите их в ответе по одному на строке. В ответе должно быть ровно шесть строк. Если не получается заполнить какой-то из пропусков, оставьте эту строку пустой.

На месте пропуска может быть либо число (в таком случае выписывайте число в десятичной системе счисления), либо одно из выражений, указанных ниже под буквами (в таком случае выписывайте соответствующую букву):

A. a
B. A
C. x
D. y
E. z
F. *a
G. *A
H. *x
I. *y
J. *z
K. x + y
L. x + z
M. x * 9
N. y * 4
O. z * 2

Пример форматирования ответа:

```
100
A
200
B
50
F
```

Задача 4-Decompile: Декомпиляция программы с циклом

Дан фрагмент кода на языке Си с тремя пропусками:

```
#define N 1024

unsigned int u[N], v[N], r[N];
unsigned int n;

...
for (unsigned i = n; пропуск_1) {
    пропуск_2 ;
    пропуск_3 ;
}
```

Для этого фрагмента (после ...) компилятор построил следующий код:

```
        mov     ecx, dword[n]
        mov     esi, 5
.S:
        test    ecx, ecx
        je     .F
        mov     ebx, dword[r + ecx*4]
        xor     edx, edx
        mov     eax, ebx
        div     esi
        test    edx, edx
        cmovne  edx, eax
        shr     ebx, 1
        mov     dword[u + ecx*4], edx
        dec     ecx
        xor     dword[v + ecx*4], ebx
        jmp     .S
.F:
```

Вам необходимо восстановить пропуски в исходном фрагменте кода на языке Си. Выпишите их в ответе по одному на строке. В ответе должно быть ровно три строки. Если не получается заполнить какой-то из пропусков, оставьте эту строку пустой.

Внимание: запрещено менять переменную `i` во 2-ом и 3-м пропусках.

Пример форматирования ответа:

```
i<=5; i*=2
n = 3
n += 7
```

Задача 5. Выражения

В волшебной стране Раз-Два-Ляндии студентов много, а компьютеров мало. Поэтому кафедра усправления вычислительными ресурсами попросила своих студентов добровольно переписать приведенные фрагменты кода на языке Си в **эквивалентные** (вычисление которых порождает одни и те же изменения в памяти для всех возможных вариантов значений переменных) фрагменты кода на языке ассемблера. К сожалению студенты, в отличие от компьютеров, допускают ошибки. Помогите их исправить, и для каждого фрагмента укажите, какие ошибки были допущены при его написании, и выпишите в ответе соответствующие буквы.

1.

```
static int a[3][4];
static short b = 0;
...
a[2][3] = b;
```

```
section .bss
    a resd 12
section .data
    b dw 0
section .text
    ...
    mov dword[a + ((2 * 4) + 3) * 4], 0
```

2.

```
static int x;
static unsigned short *p;
...
x = ++*p;
```

```
section .bss
    x: resd 1
    p: resd 1
section .text
    ...
    mov ecx, dword[p]
    inc word[ecx]
    mov cx, word[ecx]
    movsx dword[x], cx
```

3.

```
static int* a;
static int* b;
...
(*a &= 1) || (*b -= 1);
```

```
section .bss
    a resd 1
    b resd 1
section .text
    ...
    mov eax, dword[a]
    mov ecx, dword[b]
    and eax, 1
    lea edx, [.full + (.short - .full) * eax]
    jmp edx
.full:
    dec ecx
.short:
    mov dword[a], eax
    mov dword[b], ecx
```

В ответе выпишите три строки, соответствующие трем приведенным фрагментам. В каждой строке укажите **все** буквы, соответствующие допущенным в нем ошибкам. Выбирайте вариант "Другие ... ошибки ...", только если найденная вами ошибка не попадает ни под один из перечисленных вариантов. Если какой-либо тип ошибок встречается несколько раз, укажите его букву только один раз. Если ошибки отсутствуют, укажите единственную букву, соответствующую этому варианту.

A. Ошибки при объявлении переменных.
B. Некорректные инструкции/операнды.
C. Несоответствие знаковости чисел и операций над ними.
D. Ошибки в вычислениях из-за неправильного размера операндов.
E. Ошибки при применении побочных эффектов.
F. Ошибки при вычислении адресов и адресных выражений.
G. Ошибки при разыменовании указателей.
H. Другие синтаксические ошибки (ошибки использования конструкций языка).
I. Другие семантические (смысловые) ошибки.
J. Ошибки отсутствуют.

Пример правильно форматированного ответа:

```
ABC
J
EHI
```

Задача 6-Struct: Структуры и объединения

В результате взлома тестирующей системы оч. умелыми хакерами был получен фрагмент секретной неофициальной системы выставления оценок по курсу АЭиЯА. Однако восстановление типов в этом непростом случае оказалось хакерам не по зубам, и они просят Вашей помощи!

Дано определение типа структуры (с пропусками) и объявления двух глобальных переменных на языке Си:

```
struct student_scoring {
        const char *student_id;
        signed ▮▮▮ lections_visited;
        signed ▮▮▮ chocolates_gifted_per_day;
        union {
                ▮▮▮ exam_score;
                int (*test_scores)[5];
        };
        signed char times_drank_near_machines;
} *ivanov;
int ivanov_score;
```

Хакерам известно, что типы полей lections_visited и chocolates_gifted_per_day различны, а поле exam_score имеет некий вещественный тип, и что при компиляции кода в 32-разрядном окружении (неизвестно, Linux или Windows) структура в памяти выравнивается по 8-байтовой границе, а для сохранения значения некоторого поля в переменную ivanov_score был сгенерирован следующий код:

```
        mov     ebx, dword[ivanov]
        movsx   ebx, word[ebx+6]
        mov     dword[ivanov_score], ebx
```

Вас просят восстановить пропуски в типах полей структуры, а также вычислить размер структуры и смещения её полей.

I. В **первой строке** ответа заполните пропуски в типах полей `lections_visited`, `chocolates_gifted_per_day` и `exam_score`. Для этого выпишите через пробел **три буквы**, соответствующие верным вариантам:
   A. `char`
   B. `short`
   C. `int`
   D. `float`
   E. `double`
   F. `char *`
   G. `short *`
   H. `int *`
   I. `float *`
   J. `double *`
   K. `struct student_scoring *`
   L. `ни один из перечисленных типов не подходит`
   M. `подходят несколько из перечисленных типов`

II. Во **второй строке** ответа выпишите через пробел 7 чисел:
   1. смещение поля `student_id` в байтах от начала структуры,
   2. смещение поля `lections_visited` в байтах от начала структуры,
   3. смещение поля `chocolates_gifted_per_day` в байтах от начала структуры,
   4. смещение поля `exam_score` в байтах от начала структуры,
   5. смещение поля `test_scores` в байтах от начала структуры,
   6. смещение поля `times_drank_near_machines` в байтах от начала структуры,
   7. размер структуры в байтах.

Пример форматирования ответа:

```
D E C
1 2 3 4 5 6 7
```