

```

void shift_stack(long long int a, long long int b, long long int& sum)
{
    __asm
    {
        mov eax, [ebp + 8] // В eax(32 бита) записываем нижние 4 байта слова a в eax (с 8 по 11 байт)
        mov ebx, [ebp + 16] // mov записывает еще 4 байта в ebx (с 16 по 19)
        add eax, ebx // складываем
        mov esi, [ebp + 24] // Передаем в esi еще 4 байта
        mov [esi], eax // Теперь в esi храница сумма нижних частей
        mov eax, [ebp + 12] // Записываем в eax 4 байта (с 12 по 15)
        mov ebx, [ebp + 20] // Записываем в ebx 4 байта (с 20 по 23)
        adc eax, ebx // Складываем верхнюю часть
        mov [esi + 4], eax // Записываем в esi 2-е 32 битное слово учитывая смещение в 4 байта
    }
}

```

ebp

```

void shift_reg()
{
    __asm
    {
        mov eax, [ebx] // В eax кладем 4 байта ebx(a)
        mov ecx, [edx] // В ecx кладем 4 байта edx(b)
        add eax, ecx // Складываем eax и ecx
        mov [esi], eax // В esi передаем сумму eax
        mov eax, [ebx + 4] // В eax кладем следующие 4 байта ebx(a) (4-7)байт
    }
}

```

```

        mov ecx, [edx + 4] // В ecx кладем следующие 4 байта edx(a) (4-7)байт
        adc eax, ecx // Складываем eax и ecx с учетом смещения
        mov [esi + 4], eax // Добавляем остальные 4 байта в esi
    }
}
/*****/
// Main func
{
    __asm
    {
        lea ebx, a // Адрес a передаем в ebx
        lea edx, b // Адрес b передаем в edx
        lea esi, sum // Адрес sum передаем в esi
    }
    shift_reg(); // Вызываем сумму
}

```

```

long long int globA = 0;
long long int globB = 0;
long long int* globSum = 0;

void shift_global()
{
    __asm
    {
        lea ebx, globA // Кладем в ebx и edx адреса globA и globB
        lea edx, globB
        mov esi, globSum // Кладем в esi значение globSum
        mov eax, [ebx] // В eax, ecx, кладем значения ebx, edx
        mov ecx, [edx]
        add eax, ecx // Складываем сумму первых 4 байт
        mov [esi], eax // Записываем в esi первые 4 байта суммы
        mov eax, [ebx + 4] // В eax, ecx, кладем значения ebx, edx (остальные 4 байта)
        mov ecx, [edx + 4]
        adc eax, ecx // Складываем сумму последних 4 байт с учетом смещения
    }
}

```

```
        mov [esi + 4], eax // Записываем в esi последние 4 байта суммы
    }
}
/*****/
//Main func
{
    globA = a; // Передаем в globA значение a
    globB = b; // Передаем в globB значение B
    globSum = &sum; // Передаем в globSum адрес sum
    shift_global(); //Вызываем функцию
}
```