Семинар 5 - Ассемблер AArch64.

Задача

Реализуйте функцию `_print_list` на ассемблере AArch64. Функция принимает указатель на первый элемент связного списка и должна вывести в консоль содержимое полей `element` всех элементов используя функцию `puts`.

Заголовочный файл:

```
typedef struct list_entry {
  struct list_entry* next;
  const char* element;
} list_entry;

void print_list(list_entry* first);
```

Заготовка:

```
 _print_list:
  sub sp, sp, 32
  stp x29, x30, [sp, +16]
  mov x29, sp
  # <ваш код>



  # Заготовка: печать первого элемента
  ldr x0, [x0, 8] # x0 = first->element
  bl _puts        # puts(first->element)



  # </ваш код>
  ldp x29, x30, [sp, +16]
  add sp, sp, 32
  ret
```

Инструкции обработки данных

```
ADD <Rd>, <Rn>, <Rm>: Rd = Rn + Rm;
SUB <Rd>, <Rn>, <Rm>: Rd = Rn - Rm;
MUL <Rd>, <Rn>, <Rm>: Rd = Rn * Rm;
```

Инструкции load / store

```
LDR <Rt>, [<Rn>]:          Rt = *Rn;
LDR <Rt>, [<Rn>, <imm>]:   Rt = *(Rn + imm);
LDR <Rt>, [<Rn>, <imm>]!:  Rt = *(Rn += imm);
LDR <Rt>, [<Rn>], <imm>:   Rt = *(Rn); Rn += imm;
STR <Rt>, [<Rn>]:          *Rn = Rt;
STR <Rt>, [<Rn>, <imm>]:   *(Rn + imm) = Rt;
STR <Rt>, [<Rn>, <imm>]!:  *(Rn += imm) = Rt;
STR <Rt>, [<Rn>], <imm>:   *(Rn) = Rt; Rn += imm;

STP <Ra>, <Rb> -||-:       Аналогично, но сохраняет два регистра
LDP <Ra>, <Rb> -||-:       Аналогично, но загружает два регистра
```

Инструкции перехода

```
B <label>:  goto label
BL <label>: call, или goto label + r30 = <return address>
RET:        return, или goto r30
```

Логические инструкции

```
AND <Rd>, <Rn>, <Rm>: Rd = Rn & Rm;
ORR <Rd>, <Rn>, <Rm>: Rd = Rn | Rm;
EOR <Rd>, <Rn>, <Rm>: Rd = Rn ^ Rm;
```

Условные суффиксы

Чтобы изменить флаги, **добавь суффикс `S`**

`.EQ`:	Если равно	`.NE`:	Если не равно
`.HS`:	Если беззнаково $\geq$	`.LO`:	Если беззнаково $<$
`.MI`:	Если отрицательно	`.PL`:	Если положительно или ноль
`.VS`:	Если переполнение	`.VC`:	Если нет переполнения
`.HI`:	Если беззнаково $>$	`.LS`:	Если беззнаково $\leq$
`.GE`:	Если знаково $\geq$	`.LT`:	Если знаково $<$
`.GT`:	Если знаково $>$	`.LE`:	Если знаково $\leq$