2 Реализация Belt

Задача. Трент разрабатывает шифратор Amgine и решает использовать в шифраторе алгоритмы Belt (СТБ 34.101.31). Трент организовал реализацию алгоритмов на языке Си.

Алиса написала для Трента функцию

```
uint32 G(uint32 x, int r);
```

которая выполняет преобразование G_r 32-разрядного слова `x`.

Боб написал функцию, которая реализует такт зашифрования:

```
void R(uint32* a, uint32* b, uint32* c, uint32* d,
       const uint32* key, uint32 i)
{
  uint32 e;
  *b ^= G(*a + key[7 * i - 7 & 7], 5);
  *c ^= G(*d + key[7 * i - 6 & 7], 21);
  *a -= G(*b + key[7 * i - 5 & 7], 13);
  e = G(*b + *c + key[7 * i - 4 & 7], 21);
  *b += e;
  *c -= e;
  *d += G(*c + key[7 * i - 3 & 7], 13);
  *b ^= G(*a + key[7 * i - 2 & 7], 21);
  *c ^= G(*d + key[7 * i - 1 & 7], 5);
  e = *a, *a = *b, *b = *d, *d = *c, *c = e;
  e = 0;
}
```

Трента не устраивает, что в функции используется локальная переменная `e`. Ресурсы Amgine ограничены и на счету каждый байт стека.

Помогите Бобу переписать функцию так, чтобы в ней не было локальных переменных.

Решение. Вот один из вариантов решения:

```
void R(uint32* a, uint32* b, uint32* c, uint32* d,
       const uint32* key, uint32 i)
{
  *b ^= G(*a + key[7 * i - 7 & 7], 5);
  *c ^= G(*d + key[7 * i - 6 & 7], 21);
  *a -= G(*b + key[7 * i - 5 & 7], 13);
  *c += *b;
  *b += G(*c + key[7 * i - 4 & 7], 21);
  *c -= *b;
  *d += G(*c + key[7 * i - 3 & 7], 13);
  *b ^= G(*a + key[7 * i - 2 & 7], 21);
  *c ^= G(*d + key[7 * i - 1 & 7], 5);
  *a ^= *b, *b ^= *a, *a ^= *b;
  *c ^= *d, *d ^= *c, *c ^= *d;
  *b ^= *c, *c ^= *b, *b ^= *c;
}
```

□