Byte Offset Array

ZPRAC-16-17-Lab10

[BONUS]

In C an integer (int) is represented by 32 bits or 4 bytes.
For example, the integer 20 is represented in binary as:

```
00010100 00000000 00000000 00000000 (byte ordering is right to left, (little-endian))
---b3--- ---b2--- ---b1--- ---b0---
```

An array of integers is simply several such integers placed consecutively in memory.
For example, the array {12, 20} is represented in memory as:

```
{00001100 00000000 00000000 00000000} {00010100 00000000 00000000 00000000}
<----------------12-----------------><----------------20----------------->
```

Your task is to print such an array at offsets of 1,2,3 and 4 bytes (only print N-1 elements since otherwise it will overflow).
Given the above array:

1 byte offset:

```
00001100 {00000000 00000000 00000000 00010100} 00000000 00000000 00000000
     {<-------------335544320------------->}
```

2 bytes offset:

```
00001100 00000000 {00000000 00000000 00010100 00000000} 00000000 00000000
         { <--------------1310720-------------->}
```

3 bytes offset:

```
00001100 00000000 00000000 {00000000 00010100 00000000 00000000} 00000000
               {<---------------5120---------------->}
```

4 bytes offset:

```
00001100 00000000 00000000 00000000 {00010100 00000000 00000000 00000000}
                   { <----------------20----------------->}
```

INPUT FORMAT:

N (int) --- integer denoting number of elements in array

```
a1 a2 a3 ... aN --- N integers (elements of array)

OUTPUT FORMAT:
--------------
b1 b2 ... bN-1 --- N-1 integers at offset of 1 byte
c1 c2 ... cN-1 --- N-1 integers at offset of 2 bytes
d1 d2 ... dN-1 --- N-1 integers at offset of 3 bytes
e1 e2 ... eN-1 --- N-1 integers at offset of 4 bytes

EXAMPLE:
--------
INPUT:
2
12 20

OUTPUT:
335544320
1310720
5120
20
```