PRODUCT HEADER VALIDATION

Nope.

```
void __fastcall QSPI_Read_Product_Headers(struct_configuration_script_ptr *configuration_script_ptr, char *img_offset)
{
  char backup_product_header_buff[258]; // <----- FIXED BUFFER
  __int16 flash_cfg_len; // 2-byte size
  char product_header_id[4];
  unsigned __int16 product_header_length; // 2-byte size
  char is_valid; //
```

backup_product_header_buff[258] - fixed length value

product_header_length - user controlled 2-byte value in the Product Header (Length of Flash Config Section)

```
// read the entire backup header, including the stored CRC at the end
QSPI_Get_Read_Result(backup_product_header_buff, product_header_length + 2);
```

This one was trying its very best to be useful

```
product_header_length = flash_cfg_len + 0x16;
```

PRODUCT HEADER VALIDATION

```
// check the first Flash Product Header (0x0) calculated CRC against the value stored in the Flash Product Header
is_valid = QSPI_read_header_check_crc(configuration_script_ptr->flash_header_ptr, product_header_length);// <---- FAIL THIS ONE
if ( is_valid != 1 )  // if the first Flash Product Header CRC check fails, check the next at offset 0x1000 in QSPI
{
  QSPI_Cycle_CS();   // begin process to check backup Flash Product Header
  // read the flash config length from the next header in QSPI (0x1000 + 0x14)
  QSPI_Send_Read_Request(3, configuration_script_ptr->flash_header_ptr + 0x1014);
  QSPI_Get_Read_Result((char *)&flash_cfg_len, 2u);
  QSPI_reset();
  product_header_length = flash_cfg_len + 0x16;// calculate the entire length for the CRC check
  // check the second Flash Product Header (0x1000) CRC against the stored CRC value
  is_valid = QSPI_read_header_check_crc(configuration_script_ptr->flash_header_ptr + 0x1000,flash_cfg_len + 0x16);// <---- LARGE VALID HEADER
  // if the second Flash Product Header CRC fails, indicate as such and return to caller
  if ( is_valid != 1 )
  {…
  }
  // at this point the first header has failed CRC and the second has passed
  // read the backup buffer and write it to the primary location
  QSPI_Cycle_CS();
  // initiate a QSPI read at the start of the backup Flash Product Header (0x1000)
  QSPI_Send_Read_Request(3, configuration_script_ptr->flash_header_ptr + 0x1000);
  // read the entire backup header, including the stored CRC at the end
  QSPI_Get_Read_Result(backup_product_header_buff, product_header_length + 2); // I AM THE APP NOW.
  QSPI_reset();
  WDOG_feed_ff_();
  QSPI_Cycle_CS();
  QSPI_sector_erase(configuration_script_ptr->flash_header_ptr);// erase the primary product header at 0x0
  WDOG_feed_ff_();
```