Permutation inversion

Implement in assembly language a function callable from the C language:

```
bool inverse_permutation(size_t n, int *p);
```

The function takes arguments: a pointer `p` to a non-empty array of integers and the size of this array `n`. If the array pointed to by `p` contains a permutation of numbers from 0 to n-1, the function reverses this permutation in place, and the function returns true. Otherwise, the function returns false, and the content of the array pointed to by `p` after the function execution is the same as at the moment of its invocation. The function should detect obviously incorrect values of n - see the example of use. However, it is allowed to assume that the pointer `p` is valid.

Compilation

The solution will be compiled with the command:

```
nasm -f elf64 -w+all -w+error -o inverse_permutation.o inverse_permutation.asm
```

The solution must compile in the computer laboratory.

Usage example

An example of usage is provided in the file `inverse_permutation_example.c`. It can be compiled and linked with the solution using the following commands:

```
gcc -c -Wall -Wextra -std=c17 -O2 -o inverse_permutation_example.o
inverse_permutation_example.c
gcc -z noexecstack -o inverse_permutation_example inverse_permutation_example.o
inverse_permutation.o
```