permgen user manual

Title	permgen (Permutation generator)
Author	Nikolaos Kavvadias
Contact	nikolaos.kavvadias@gmail.com
Website	http://www.nkavvadias.com
Release Date	08 July 2017
Version	0.0.2
Rev. history	
v0.0.2	2017-07-08 Add README, LICENSE, Makefile. Create github project.
v0.0.1	2013-03-03 Initial release.

1. Introduction

`permgen` is a permutation generator writte in ANSI/ISO C. Most algorithms are based on:

Donald E. Knuth, "The Art of Computer Programming, volume 4,
fascicle 2B, Chapter 7.2.1.2: Generating all permutations".

2. File listing

The `permgen` distribution includes the following files:

/permgen	Top-level directory
LICENSE	Description of the Modified BSD license.
Makefile	Makefile for generating the `permgen` executable.
README.html	HTML version of README.rst.
README.pdf	PDF version of README.rst.
README.rst	This file.
permgen.c	The source code for the application.
rst2docs.sh	Bash script for generating the HTML and PDF versions.
test1.gap	Example set/multiset description file.
test.sh	Bash script for testing the application.

3. Installation

There exists a quite portable Makefile (`Makefile` in the current directory). Running `make` from the command prompt should compile `permgen`.

To enable easier hardware compilation, the user can select to perform static allocations and disable any array interfaces.

- `NOMALLOC`: use statically-allocated arrays; no `malloc()` and `free()`.
- `NOARRIF`: no function argument array interfaces.

4. Prerequisites

- [mandatory for building] Standard UNIX-based tools * host compiler (e.g., `gcc`) * `make` * `bash`

5. permgen usage

`permgen` can be invoked as follows:

```
$./permgen [options]
```

The complete `permgen` options listing:

-h Print this help.

-l Use algorithm L (lexicographic permutation generation). This is the default generator.

-p Use algorithm P (plain changes method)

-c Use algorithm C (permutation generation by cyclic shifts).

-e Use algorithm E (permutation generation by Ehrlich swaps).

-i <file> Specify input file with set/multiset elements. The input file for this program holds the values of the elements of the set, for which all permutations will be generated.

Here is a sample input file (note that 3 is missing and 2 appears twice): 1 2 2 4 5

-o <file> Specify output file.

-n <num> Specify number of increasing elements in a set. This option is used if no input file is specified.

Here follow some simple usage examples of `permgen`.

1. Use algorithm L to generate all permutations for the set/multiset description in file `test1.gap` and write the results to `perms.txt`.

```
$ ./permgen -l -i test1.gap -o perms.txt
```

2. Generate all the permutations for a 3-element set {1, 2, 3} and write the results to file `perms123.txt`.

```
$ ././permgen.exe -n 3 -o perms123.txt
```

6. Running tests

In order to run a series of sample tests do the following:

```
$ ./test.sh
```