

Project	IEEE 802.16 Broadband Wireless Access Working Group	
Title	Summary of MAC and PHY proposal evaluation results at session #4	
Date Submitted	1999-11-14	
Source	Brian Petry 3Com 12230 World Trade Dr. San Diego, CA 92128	Voice: 858-674-8533 Fax: E-mail: brian_petry@3com.com
Re:	At 802.16 session #4, the working group undertook a proposal evaluation scoring process for MAC and PHY proposals, as described in 802.16-99/07. The proposal evaluation is part of the 802.16.1 standard development plan, 802.16-99/05. The current document describes the results of MAC and PHY scoring which took place at session #4. Brian Petry, appointed to an ad-hoc score compilation committee, collected score cards from 802.16 voters. These score cards were output from MS-Excel spreadsheets. Brian ran the score cards through a program that calculated the results described herein.	
Abstract	The scoring results show the proposals meeting and failing the evaluation criteria. Also shown are some statistical results for each proposal and each criterium: min, max, average (mean) and standard deviation. The current document, which to date has been made publicly available, reports the results and statistics, but not voters' score cards. Another document, which to date has been made available to 802.16 voting members only, reports the score cards from each voter. In an appendix, the score compilation program code is included.	
Purpose	The purpose of this document is to inform voting members of the results of proposal evaluation scoring. Proposers may use this information in the next cycle of proposal evaluation to help revise their proposals, merge with other proposers, etc.	
Notice	This document has been prepared to assist the IEEE 802.16. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein.	
Release	The contributor acknowledges and accepts that this contribution may be made public by 802.16.	
IEEE Patent Policy	The contributor is familiar with the IEEE Patent Policy, which is set forth in the IEEE-SA Standards Board Bylaws < http://standards.ieee.org/guides/bylaws > and includes the statement: "IEEE standards may include the known use of patent(s), including patent applications, if there is technical justification in the opinion of the standards-developing committee and provided the IEEE receives assurance from the patent holder that it will license applicants under reasonable terms and conditions for the purpose of implementing the standard."	

Report of MAC and PHY proposal evaluation results at session #4

Brian Petry

1 MAC Results

For help interpreting the proposal numbers and criteria numbers, please use the following links:

Criteria: <http://grouper.ieee.org/groups/802/16/meetings/mtg04/mac.html>

Proposals: http://grouper.ieee.org/groups/802/16/mac/docs/80216m-99_01.html

Proposals meeting the criteria:

04
05
07
08
09
10
11
13
14
16
17

Proposals not meeting the criteria:

06
12
15

31 ballots processed

1.1 Statistics

Minimum Scores

	1	2	3	4	5	6	7	8	9	10	11	12
04:	2.0	2.0	1.9	2.2	2.0	2.0	2.3	2.0	1.0	3.0	4.0	2.0
05:	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.4	0.3	0.5
06:	0.5	0.0	0.5	0.0	0.0	0.0	0.0	0.5	0.5	0.0	0.0	0.0
07:	1.1	2.0	2.0	3.0	2.1	1.1	3.0	3.0	2.0	5.9	3.1	2.5
08:	3.0	3.0	3.0	2.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0
09:	1.0	1.0	2.0	2.0	2.0	1.0	2.0	1.0	2.0	2.2	2.2	2.0
10:	3.0	3.0	3.0	1.0	3.0	3.0	3.0	3.0	3.0	1.3	3.0	3.0
11:	0.0	4.0	2.0	2.0	3.0	3.0	0.5	0.3	0.0	0.0	0.0	2.0
12:	0.5	0.0	0.5	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0
13:	3.0	3.0	1.9	2.2	2.0	3.0	0.5	0.3	1.0	3.0	3.0	2.0
14:	1.1	2.0	1.0	2.0	2.0	1.0	2.0	0.3	1.0	2.2	2.2	2.5
15:	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
16:	0.5	2.0	1.9	2.2	3.0	1.1	3.0	3.0	2.0	3.0	3.0	3.0
17:	4.0	4.0	4.0	4.0	4.0	4.0	4.0	3.0	4.0	2.0	3.0	3.0

Maximum Scores

	1	2	3	4	5	6	7	8	9	10	11	12
04:	10.0	8.0	9.0	10.0	9.0	9.0	9.0	10.0	9.0	10.0	9.7	9.0
05:	10.0	9.0	10.0	8.0	9.0	10.0	10.0	10.0	10.0	9.0	8.0	10.0
06:	9.0	7.0	8.0	8.0	9.9	9.5	10.0	7.3	10.0	8.0	8.4	10.0

07:	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
08:	10.0	10.0	9.9	9.6	9.9	10.0	9.0	10.0	9.0	9.0	8.0	10.0
09:	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
10:	10.0	10.0	10.0	10.0	10.0	10.0	10.0	9.7	10.0	10.0	10.0	10.0
11:	8.0	10.0	9.4	8.0	9.0	9.0	8.0	8.0	9.0	8.0	8.0	9.0
12:	8.0	7.0	7.6	8.0	7.9	8.0	8.0	8.0	7.0	7.0	7.8	8.0
13:	9.0	8.0	9.0	9.0	8.0	10.0	9.0	10.0	9.0	9.0	9.0	9.0
14:	8.3	10.0	8.0	8.0	8.8	8.0	8.0	8.0	8.0	9.1	9.0	10.0
15:	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
16:	9.0	9.0	9.0	9.0	9.0	10.0	9.0	10.0	9.0	9.0	9.0	9.0
17:	8.3	8.0	8.0	8.0	8.8	8.5	7.7	8.0	7.0	8.0	8.0	8.0

Standard Deviation

	1	2	3	4	5	6	7	8	9	10	11	12
04:	2.1	1.6	2.1	2.0	2.1	1.8	1.7	1.8	2.2	1.8	1.5	1.9
05:	2.3	2.0	2.2	1.8	1.9	2.0	2.3	2.2	2.1	2.1	2.0	2.5
06:	2.5	2.1	2.1	2.1	2.4	2.3	2.3	2.1	2.1	2.0	2.2	2.2
07:	2.1	2.0	2.2	2.0	1.7	2.1	1.5	1.7	2.0	1.2	1.5	1.7
08:	1.7	1.6	1.6	1.6	1.6	1.8	1.8	1.8	1.4	1.2	1.1	1.8
09:	2.2	1.9	1.8	1.8	1.7	2.1	1.4	1.8	1.5	1.6	1.5	1.7
10:	1.5	1.7	1.8	2.1	1.9	1.6	1.6	1.7	1.6	1.8	1.6	1.6
11:	2.2	1.7	2.0	1.6	1.6	1.9	1.7	2.2	2.2	2.1	2.1	1.7
12:	1.8	1.7	1.7	1.6	1.8	1.9	1.9	2.0	1.9	1.7	1.7	1.8
13:	1.6	1.7	2.0	1.6	1.8	1.6	1.7	2.1	1.9	1.5	1.2	1.6
14:	2.0	1.8	1.5	1.6	1.4	2.0	1.3	1.8	1.5	1.6	1.5	1.5
15:	2.6	2.5	2.3	2.4	2.4	2.6	2.3	2.1	2.3	2.3	2.4	2.2
16:	1.9	1.9	1.8	1.8	1.9	2.0	1.4	1.8	2.0	1.3	1.4	1.6
17:	1.1	1.1	1.0	1.0	1.1	1.2	0.9	1.2	1.0	1.5	1.3	1.1

Average Scores

	1	2	3	4	5	6	7	8	9	10	11	12	Avg
04:	6.2	5.9	5.9	6.3	5.8	6.0	6.2	6.5	6.3	6.8	6.8	5.9	6.2
05:	6.1	5.4	5.4	5.5	5.7	5.6	5.8	5.3	5.7	5.1	5.5	6.0	5.6
*06:	5.3	4.3	4.9	5.0	5.2	5.0	5.1	4.7	5.0	4.9	5.3	5.1	5.0
07:	6.5	6.9	6.1	6.7	6.5	6.6	6.3	6.7	6.6	7.5	7.1	6.6	6.7
08:	6.7	6.5	6.6	6.0	6.3	7.0	6.4	6.6	6.1	5.8	6.1	6.5	6.4
09:	6.2	6.2	5.6	6.0	6.1	6.0	5.9	6.0	6.2	6.8	6.5	5.9	6.1
10:	7.4	6.8	6.7	6.0	6.5	7.1	6.6	6.5	6.8	6.0	6.7	6.8	6.7
11:	5.2	7.0	6.0	5.6	5.9	6.3	5.4	4.5	5.1	4.9	5.1	5.9	5.6
*12:	5.3	5.0	5.2	5.3	5.2	5.3	4.9	5.2	4.8	5.0	5.6	5.5	5.2
13:	6.6	5.9	6.0	6.2	5.9	6.4	5.9	6.3	5.9	6.3	6.5	5.8	6.1
14:	5.9	5.7	5.2	5.4	5.9	5.7	5.7	5.9	5.7	6.3	6.0	6.0	5.8
*15:	5.1	5.2	5.1	5.1	5.2	5.1	5.2	5.4	5.1	5.5	5.5	5.0	5.2
16:	6.6	6.1	6.2	6.5	6.3	6.5	6.5	6.9	6.2	7.0	6.9	6.1	6.5
17:	5.9	5.9	5.8	5.8	5.8	6.0	5.8	5.9	5.7	5.4	5.9	5.9	5.8

2 PHY Results

For help interpreting the proposal numbers and criteria numbers, please use the following links:

Criteria: <http://grouper.ieee.org/groups/802/16/meetings/mtg04/phy.html>

Proposals: http://grouper.ieee.org/groups/802/16/mac/docs/80216p-99_01.html

Proposals meeting the criteria:

7

9

10

11
12
16
17
19
20
21
23
25

Proposals not meeting the criteria:

8
13
14
18&24
22
27

32 ballots processed

2.1 Statistics

Minimum Scores

	1	2	3	4	5	6	7	8	9	10	11
7:	3.0	4.0	4.0	1.0	3.0	3.0	3.0	3.0	3.0	4.0	4.0
8:	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
9:	0.0	4.0	4.0	5.0	4.0	3.0	4.0	2.0	4.0	2.0	3.0
10:	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11:	0.0	4.0	4.0	3.0	3.0	3.0	4.0	2.0	4.0	4.0	3.0
12:	3.0	4.0	4.0	3.0	3.0	4.0	4.0	4.0	5.0	5.0	3.0
13:	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	1.0	1.0
14:	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0
16:	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.5	0.5
17:	3.0	4.0	5.0	3.0	3.0	4.0	4.0	4.0	5.0	5.0	3.0
18&24:	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
19:	0.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0
20:	0.4	3.0	3.0	3.0	3.0	3.0	0.0	0.0	3.0	2.0	0.0
21:	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0
22:	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.3	1.0	1.0
23:	0.4	2.0	1.0	2.0	0.0	2.0	0.0	0.0	2.0	2.0	2.0
25:	2.0	3.0	1.0	3.0	4.0	2.0	2.0	4.0	2.0	1.0	3.0
27:	3.0	3.0	1.0	3.0	3.0	3.0	3.0	1.0	3.0	1.0	3.0

Maximum Scores

	1	2	3	4	5	6	7	8	9	10	11
7:	10.0	9.0	10.0	9.0	9.0	9.0	10.0	10.0	9.0	9.0	9.0
8:	9.0	9.0	8.0	8.0	9.0	8.0	8.0	8.0	9.0	9.0	8.0
9:	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
10:	9.3	9.0	9.0	10.0	9.5	9.4	9.7	9.1	9.9	9.3	9.3
11:	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
12:	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
13:	10.0	8.0	8.0	9.0	9.0	7.5	6.0	6.0	7.0	8.5	8.5
14:	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0
16:	9.0	9.0	8.0	10.0	9.0	9.0	9.0	9.0	9.0	9.0	9.5
17:	10.0	10.0	10.0	9.0	10.0	10.0	10.0	10.0	9.0	9.0	9.0
18&24:	7.0	7.0	6.0	6.0	7.0	7.0	8.0	8.0	7.0	8.0	7.8
19:	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	9.0
20:	8.0	10.0	8.0	10.0	8.0	8.0	8.0	9.0	10.0	8.0	8.0
21:	8.6	9.0	9.0	9.0	9.0	9.0	10.0	10.0	9.0	9.0	9.0

22:	8.0	9.0	9.0	9.0	10.0	8.0	8.0	8.0	8.0	8.0	8.0
23:	8.6	9.0	9.0	9.0	9.0	9.0	9.0	9.0	10.0	10.0	10.0
25:	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	9.7	10.0	10.0
27:	8.0	8.0	8.0	7.0	8.0	8.0	8.0	7.0	8.0	9.0	8.0

Standard Deviation

	1	2	3	4	5	6	7	8	9	10	11
7:	1.5	1.6	1.7	1.9	1.7	1.7	1.9	1.6	1.5	1.4	1.5
8:	1.9	2.1	1.5	1.7	2.0	1.6	2.0	1.7	1.7	1.7	1.6
9:	2.2	1.7	1.3	1.3	1.5	1.5	1.6	2.0	1.4	1.6	1.7
10:	2.0	2.0	1.9	2.2	2.0	1.9	1.9	1.8	2.2	2.3	2.2
11:	2.3	1.4	1.4	1.7	1.7	1.7	1.6	1.8	1.4	1.3	1.4
12:	1.4	1.3	1.2	1.3	1.5	1.3	1.4	1.4	1.1	1.2	1.4
13:	2.1	1.7	1.6	1.7	1.8	1.8	2.2	2.1	2.2	2.2	2.3
14:	2.2	1.8	1.6	1.6	1.7	1.6	1.5	1.8	1.5	1.7	1.5
16:	2.7	2.0	2.5	2.6	2.3	2.3	2.1	2.2	1.9	2.0	2.1
17:	1.2	1.5	1.1	1.4	1.5	1.3	1.2	1.2	1.0	1.0	1.1
18&24:	2.3	2.3	2.2	2.2	2.2	2.2	2.2	2.3	2.2	2.3	2.2
19:	2.4	2.0	1.8	1.8	2.1	2.1	2.1	2.0	1.9	1.9	2.0
20:	1.7	1.5	1.1	1.4	1.2	1.3	1.7	1.9	1.4	1.4	1.7
21:	1.4	1.4	1.3	1.4	1.7	1.6	1.7	1.5	1.1	1.3	1.5
22:	2.2	1.7	1.9	1.8	2.1	1.5	1.5	1.5	2.1	1.8	1.6
23:	2.0	1.6	2.2	1.7	2.3	1.7	2.3	2.6	1.7	1.8	1.8
25:	1.6	1.5	1.9	1.6	1.5	1.5	1.7	1.2	1.7	1.7	1.4
27:	1.6	1.4	1.5	1.2	1.4	1.5	1.5	1.4	1.4	1.7	1.4

Average Scores

	1	2	3	4	5	6	7	8	9	10	11	Avg
7:	7.2	6.7	6.7	6.1	6.5	6.2	6.5	6.5	6.5	6.8	6.7	6.6
*8:	5.5	5.9	5.4	5.4	5.1	5.2	5.6	5.1	5.4	5.4	5.3	5.4
9:	6.6	6.5	7.1	6.9	6.6	6.1	7.1	6.4	7.2	6.9	6.7	6.7
10:	6.3	6.1	6.1	6.5	6.1	6.1	6.0	5.9	6.5	6.6	6.5	6.2
11:	6.2	6.6	7.0	6.6	6.3	6.3	6.5	6.3	6.8	6.8	6.6	6.6
12:	6.9	6.5	6.4	6.2	6.2	6.2	6.2	6.4	6.5	6.5	6.2	6.4
*13:	5.7	5.7	5.3	5.5	5.6	5.2	4.4	4.4	4.6	5.8	5.7	5.3
*14:	5.5	5.2	5.2	5.3	5.4	5.5	5.3	5.1	5.8	5.7	5.6	5.4
16:	5.2	5.8	4.8	4.7	5.6	5.3	5.3	5.5	5.5	6.3	6.5	5.5
17:	6.8	6.7	6.8	6.3	6.6	6.3	6.2	6.6	6.6	6.7	6.4	6.5
*18&24:	3.8	4.1	3.7	3.6	3.8	3.8	3.8	3.9	4.0	3.9	4.0	3.9
19:	4.4	4.3	4.5	4.6	4.4	4.7	4.5	4.6	5.0	5.5	6.2	4.8
20:	5.9	6.1	5.7	6.4	5.9	5.8	5.3	5.6	6.4	6.0	5.9	5.9
21:	6.4	6.6	6.7	6.4	6.5	6.3	6.4	6.6	6.4	6.5	6.4	6.5
*22:	5.4	5.5	5.5	5.6	5.6	5.5	5.6	5.6	5.3	5.2	5.5	5.5
23:	5.7	6.5	5.4	5.5	5.7	6.2	5.7	5.5	6.7	6.6	6.6	6.0
25:	7.2	6.9	5.9	6.4	6.8	6.7	6.4	6.5	6.7	6.4	6.4	6.6
*27:	5.6	5.6	5.3	5.4	5.7	5.2	5.7	5.3	5.9	5.6	5.6	5.5

3 Appendix: Score Compilation Program Code

The following is the program used to process voters' score cards, calculate statistics and display results. It was written by Brian Petry and is included here with a "public domain" copyright (which is a slightly modified version of the well-known University of California copyright used for it's variants of UNIX):

```
#!/usr/local/bin/perl

# Copyright c 1999, 3Com Corporation. All Rights Reserved.
#
# Redistribution and use in source form only, with or without modification,
```

```

# is permitted provided that the following conditions are met:
#
# 1. Redistribution of the code must retain the above copyright notice, this
#    list of conditions and the following disclaimer.
# 2. All advertising materials mentioning features or use of this software
#    must display the following acknowledgement: This product includes
#    software developed by 3Com Corporation.
# 3. The name, 3Com Corporation, may not be used to endorse or promote
#    products derived from this software without specific prior written
#    permission.
#
# This software is provided by 3Com "as is" and any express or implied
# warranties, including, but not limited to, the implied warranties of
# merchantability and fitness for a particular purpose are disclaimed.
# In no event shall 3Com be liable for any direct, indirect, incidental,
# special, exemplary, or consequential damages (including, but not limited
# to, procurment of substitute goods or services; loss of use, data, or
# profits; or business interruption) however caused and on any theory of
# liability, whether in contract, strict liability, or tort (including
# negligence or otherwise) arising in any way out of the use of this software,
# even if advised of the possibility of such damage.
#####

#
# Process a bunch of csv (comma separated version) files that are output from
# Micro$oft Excel spreadsheets (saveas->csv). Each spreadsheet is a scorecard
# Each row is a proposal and each column is a category. Each csv file is
# specified on the command line.
#
#
# usage:
# perl score.pl file1 file2 ...
#
# some constants
$ignorehead = 6;          # number of lines to ignore in the header
$nmaccategories = 12;    # number of MAC score categories
$nphycategories = 11;    # number of PHY score categories
$nmacprops = 14;         # number of MAC proposals
$nphyprops = 18;         # number of PHY proposals
$pass = 6;               # avg score needed to pass

# declare some arrays
my @avgscores;           # average scores
my @sumscores;           # summed scores (used for average and std. deviation)
my @minscores;           # minimum scores
my @maxscores;           # maximum scores
my @devscores;           # standard deviation

# print a header: categories
sub header {
    print " ";
    for ($cat = 0; $cat < $ncategories; $cat++) {
        printf "%4d ", $cat+1;
    }
    if (scalar(@_) > 0) {
        print " Avg";
    }
}

```

```

    }
    print "\n";
}

if (scalar(@ARGV) == 0) {
    print "usage: perl score.pl file1 file2 ... \n";
    exit;
}

# initialize minimum scores
for ($prop = 0; $prop < 99; $prop++) {
    for ($cat = 0; $cat < 99; $cat++) {
        $minscores[$prop][$cat] = 99;
    }
}

# For each csv file specified on the command line
$voters = 0;           # accumulator: count of voters (csv files)
$phy = 0;
foreach $vfilename (@ARGV) {

    $voters++;

    open(vfile, $vfilename) or die "Can't open file $vfilename: $!\n";

    # ignore first $ignorehead lines of file
    $x = 0;
    while ($x < $ignorehead) {
        $_ = <vfile>;
        if ((split)[0] eq "PHY") {
            $phy = 1;
        }
        $x++;
    }

    # if this is a PHY score card
    if ($phy) {
        $nprops = $nphyprops;
        $ncategories = $nphycategories;
    } else {
        $nprops = $nmacprops;
        $ncategories = $nmaccategories;
    }

    # read in the scores
    for ($prop = 0; $prop < $nprops; $prop++) {
        @row = split /,/, <vfile>; # convert input row to an array

        # delete the first and extra elements of the row
        @cname[$prop] = shift @row; #save proposal name/number
        while (scalar(@row) > $ncategories) {
            pop @row;
        }

        @scores[$prop] = [ @row ]; # store the row in a 2-dimensional array

        for ($cat = 0; $cat < $ncategories; $cat++) {
            # save minimum score

```

```

    $thisscore = $scores[$prop][$cat];

    # validate that score is within range
    if ($thisscore < 0) {
        $thisscore = 0;
        $scores[$prop][$cat] = 0;
    }
    if ($thisscore > 10) {
        $thisscore = 10;
        $scores[$prop][$cat] = 10;
    }

    if ($thisscore < $minscores[$prop][$cat]) {
        $minscores[$prop][$cat] = $thisscore;
    }

    # save maximum score
    if ($thisscore > $maxscores[$prop][$cat]) {
        $maxscores[$prop][$cat] = $thisscore;
    }

    # accumulate total for mean and std dev
    $sumscores[$prop][$cat] += $thisscore;

    # accumulate sum-of-squares for standard deviation
    $devscores[$prop][$cat] += $thisscore * $thisscore;
}
}

# find the voter's name
while (<vfile>) {
    if (/family/) {
        $family = (split /,/)[0];
    }
    if (/given/) {
        $given = (split /,/)[0];
    }
}

# print, in abbreviated form, the voter's scorecard
print "$given $family: \n";
header;
for ($prop = 0; $prop < $nprops; $prop++) {
    printf "%5s:", @cname[$prop];
    for ($cat = 0; $cat < $ncategories; $cat++) {
        printf "%4.1f ", $scores[$prop][$cat];
    }
    print "\n";
}

close(vfile);
}

# calculate the average scores (needed for standard deviation also)
for ($prop = 0; $prop < $nprops; $prop++) {
    for ($cat = 0; $cat < $ncategories; $cat++) {
        $avgscores[$prop][$cat] = $sumscores[$prop][$cat]/$voters;
    }
}

```



```

}

print "\n-----RESULTS-----\n\n";

# print the minimum scores
print "$voters ballots processed\n\n";
print "Minimum Scores\n";
header;
for ($prop = 0; $prop < $nprops; $prop++) {
    printf "%5s:", @cname[$prop];
    for ($cat = 0; $cat < $ncategories; $cat++) {
        printf "%4.1f ", $minscores[$prop][$cat];
    }
    print "\n";
}

# print the maximum scores
print "Maximum Scores\n";
header;
for ($prop = 0; $prop < $nprops; $prop++) {
    printf "%5s:", @cname[$prop];
    for ($cat = 0; $cat < $ncategories; $cat++) {
        printf "%4.1f ", $maxscores[$prop][$cat];
    }
    print "\n";
}

# print the standard deviation
print "Standard Deviation\n";
header;
for ($prop = 0; $prop < $nprops; $prop++) {
    printf "%5s:", @cname[$prop];
    for ($cat = 0; $cat < $ncategories; $cat++) {
        printf "%4.1f ", sqrt(($devscores[$prop][$cat] - $avgscores[$prop][$cat] *
$sumscores[$prop][$cat]) / ($voters - 1));
    }
    print "\n";
}

# print the averaged results
print "Average Scores\n";
header(1);
for ($prop = 0; $prop < $nprops; $prop++) {
    $sum = 0;
    @passing[$prop] = 0;
    printf "%5s:", @cname[$prop];
    for ($cat = 0; $cat < $ncategories; $cat++) {
        $score = $avgscores[$prop][$cat];
        if ($score >= $pass) {
            @passing[$prop] = 1;
        }
        $sum += $score;
        printf "%4.1f ", $score;
    }
    @avg[$prop] = $sum/$ncategories;
    printf "%4.1f\n", @avg[$prop]; # print the average of all categories
}

```

```
# print proposals, sorted by rank, high score first
# create a list of proposal indices
#for ($prop = 0; $prop < $nprops; $prop++) { push @plist, $prop; }
#sub decreasing {
#  @avg[$b] <=> @avg[$a];
#}
#
#@sorted = sort decreasing @plist;
#foreach $prop (@sorted) {
#  printf "%5s:%4.1f\n", @cname[$prop], @avg[$prop];
#}

# print the proposals that pass
print "Proposals meeting the criteria:\n";
for ($prop = 0; $prop < $nprops; $prop++) {
  if (@passing[$prop]) {
    print "@cname[$prop]\n";
  }
  else {
    $failing .= "@cname[$prop]\n";
  }
}

#print the proposals that failed
print "Proposals not meeting the criteria:\n$failing";
```