

# Applying the Top-down Approach to Beginners in Programming Language Education

Daisuke Saito

Information and Telecommunication Engineering Dept.  
Tokai University  
Tokyo, Japan  
3bjnm007@mail.tokai-u.jp

Tsuneo Yamaura

Information and Telecommunication Engineering Dept.  
Tokai University  
Tokyo, Japan  
yamaura@keyaki.cc.u-tokai.ac.jp

**Abstract**—There are two approaches in learning programming language: the bottom up approach and the top-down approach. The bottom up approach starts from basic learning and shifts to real programming via data definition, and grammar. The top-down approach, on the other hand, is a process to self-learn the programming language with using formal sample program. The top down approach is more effective and requires less time to learn compared with the bottom up, but is supposed for a person who has already learned one language or more. We, however, believe that the top down approach is applicable to the first-time learners. The top down approach also fosters the understanding of reusability.

**Keywords**—*Programing language learning; Education method, Top-down Approach; Bottom-up Approach*

## I. INTRODUCTION (Heading 1)

This is quite difficult to teach a beginner a programming language and has been considered to be a long time problem [1]. It is extremely important for the programming language learners besides to fully understand the language grammar as well as how to implement quality in the source code.

Here we immediately define two of basic approaches in learning programming languages: the bottom up approach (hereafter we denote BUA) and the top-down approach (TDA): BUA is a method firstly to learn the grammar, data definition, and how to program based on the reference books and textbooks, then shifts to the actual software development. This is easy to apply to the beginners of programming languages. Another approach, TDA, is a way to continue to learn the basics and applications of programming from the abstract part of the programming language [2].

In this paper, we prove that TDA is applicable to a novice learner of a programming language, and we propose a process for improving the novice programmer's quality of the source code.

## II. TOP-DOWN APPROACH

TDA learning discussed in this paper is an extension of [3], and is a method of learning programming skills by using the sample program. Our learning method is a method that the

learners go through the learning by reading the comments attached to the source code of the sample program.

Figure 1 shows an example of a sample program.

```
//variable A.
var A
//Substitute the 4 as the initial value to the A
//Tip: Try to change the value of A
A=4
//if statement
//if (condition){ When run that matches the conditions}
//else{To execute when the condition is not met}
//Determine the even and odd
//Determine the value that is assigned to A
if(A%2==0)
{
  //on the screen, (numeric value you assign is displayed on the A) A display and is even
  document.write(A + " is even. ");
}
else
{
  //on the screen, (numeric value you assign is displayed on the A) A display and is odd
  document.write(A + " is odd. ");
}
document.write("<br>"); //line feed
document.write("<br>"); //line feed
```

Fig. 1. Sample Program

The sample program, shown in Figure 1 is a simple application program that determines whether the given number is odd or even. The comments attached to the program explain how the program works, as well as where, how, why of the program changes if there were some changes in the source code. Learners are able to understand the programming in the self-learning manner by reading the sample program. Such detailed comments help learners understand and study complex sample programs.

The following steps show the learning method using the sample program:

1. Read the comments (statements as well) of the sample source code, and try to understand the program.
2. Run the sample program, to understand the grammar and behavior of the program.
3. Change some parts of the sample program, and execute the modified program to confirm it runs as expected.

4. Use the sample program in copy-and-paste manner in the actual program development.

We can expect that the learners can enjoy the following effects from the above processes

- TDA is suitable for learners for the first time.
- TDA is more effective than BUA.
- TDA teaches the basic idea of reusability.

### III. DETAILS OF THE EXPERIMENTS

We applied our approach to 41 high school students to confirm the effect of the learning using a sample program.

#### A. Overview

##### 1) Targets

Forty-one high school students

##### 2) Number of Groups

We split the students into 2 groups for comparison with the effect of BUA and TDA. The BUA group has 21 persons and 20 for TDA

##### 3) Used Programming Language

We used JavaScript. This is because we believe JavaScript is suitable for program language learners; it is relatively easy to provide programming environment, does not require a compiler, and is a browser-based language that is easy to run GUI.

##### 4) Time Spent for the Experiments

Learning time is 120 minutes, and Problem-solving-time (coding) is 30 minutes. Total time is 150 minutes.

##### 5) Targetted Sentences to Be Learned

- Character display on the screen
- Arithmetic calculations
- Conditional branch ("if" Sentences)
- Loop ("for" Sentence)

#### B. Comparison of Two Groups

In this section, we are showing the difference in the course of BUA Group and TDA Group.

##### 1) BUA Group

- We used the reference book of JavaScript
  - A high school teacher gave a programming lecture to the students.
  - A teaching assistant supported the lecture.
- ##### 2) TDA Group
- We used seven sample programs of JavaScript.
  - The learners self-learned by reading the samples.
  - A high school teacher and a TA supported the learners.

#### C. Problem-Solving (Coding)

After the learning of 120 minutes, we gave 30 minutes to solve 3 problems. The given problems are related to a rectangle as specified below:

Problem1: Enter the length of 2 sides, and calculate the area.

Problem2: Determine if the shape is rectangular or square.

Problem 3 : Draw a rectangle with using "■" which is a 1 cm by 1cm square.

With the discussion with the high school teacher, the 3 problems were chosen and modified so that the problems fit the learners' knowledge. The learners were allowed to refer to the programs that they made during the lectures.

We showed the learners the below diagram as an example of the execution result:

```
/*problem1 Area of Quadrangle*/
Length=10
Side=10
Area of Quadrangle 100 .

/*problem2 Judge of Quadrangle*/
Square

/*problem3 Draw Quadrangle*/
■■■■■■■■■■
■■■■■■■■■■
■■■■■■■■■■
■■■■■■■■■■
■■■■■■■■■■
■■■■■■■■■■
■■■■■■■■■■
■■■■■■■■■■
■■■■■■■■■■
■■■■■■■■■■
```

Fig. 2. Examples of an Execution Result of the Problems

As far as a coded program outputted as specified (if not exactly same as the showed example), we assumed as a correct program.

We only asked only JavaScript programs: the learners received HTML for executing the problems.

### IV. RESULTS AND ANALYSIS

In this chapter, we show the results and analysis of the problems of both groups.

#### A. Results of the Problems

We attached the results of the problem of the learners of BUA (Appendix 1) and TDA (Appendix 2).

Table 1 illustrates the summary of the correct coding rate.

TABLE I. CORRECTLY CODED RATES

	TDA	BUA
Problem1	6 Per	8 Per
Problem2	1 Per	1 Per
Problem3	0 Per	0 Per

Table 1 shows that for the Problem 1, 8 students out of 21 (38.1%) were correct in the BUA group, and in the TDA group 6 out of 20 (30%). For the problem 2, in the BUA group 1 out of 21 (4.8%) was correct, in the TDA group 1 student out of 20 (5%). As for the Problem 3, none of the students in both groups could correctly perform. In terms of the “correct rate,” BUA seems to be slightly better.

### B. Analysis of the Source Code

We analyzed the source code of the Problem 1 on which all the learners (41 people) worked on. We put emphasis on the readability of the source code: we took a close look at a blank line and indentation. We referred to [4] as coding criteria.

#### 1) Indentation

We analyzed if there was an indentation in the source code of the problem 1. Table 2 shows the results. In our experiments, our definition of indentation is if there is at least one or more spaces. The data in the table includes the program that did not run correctly but had indentation.

TABLE II. DETAIL OF INDETAITION

Detail of indent	BUA	TDA
4 spaces or more (including Tab character)	0	5
less than 3 spaces	3	0

Generally speaking the indentation defined in the coding conventions is 4 spaces or more (or tabs) [5]. In the TDA Group, 5 people out of 20 (25%) indented 4 or more spaces (or, tabs). In the BUA group, 3 out of 21 people (14.3%) indented, and all of 3 students indented 3 spaces or less.

Lectures and TA's of both groups did not mention about indentation. In the TDA group, students unintentionally learned the proper sense of the indentation by reading the sample programs. In the BUA group, many students indented with one space. This may be because of BUA, but we cannot deny the possibilities of coincident.

A correct indentation in a program drastically improves the readability, maintainability, as well as the quality of the source code. TDA can teach the sense of indentation unintentionally.

#### 2) Blank line

Blank lines are also important to increase the readability of a program. We counted the LOC, or lines of code to analyze if there are blank lines. We only targeted the programs that ran correctly. Table 3 shows the LOC's of BUA, and Table 4 is for those of TDA.

TABLE III. LOC OF BUA

	BU 7	BU 8	BU 11	BU 13	BU 14	BU 15	BU 16	BU 20
LOC	7	8	7	7	7	7	7	9

TABLE IV. LOC OF TDA

	TD 4	TD 7	TD 8	TD 16	TD 17	TD 19
LOC	12	8	11	9	9	8

Average of LOC of BUA group was 7.4 lines while the TDA showed 9.5 lines. The difference of approximately 2.1 lines. We can mention that LOC of the TDA group tends to increase. This difference comes from blank lines.

The students of the TDA group follow the programming style of the sample programs, and unconsciously put blank lines.

### C. Execution Results

In this section, we analyzed the execution results of the learners who wrote correct programs for the problem 1.

We showed Figure 3 as an example the results of the Problem 1.

```

/*problem1 Area of Quadrangle*/
Length=10
Side=10
Area of Quadrangle 100 .

```

Fig. 3. Example of the Execution Result of Problem 1

We analyzed the results of 14 students (6 in TDA, 8 in BUA) who correctly made programs for the Problem 1, and graded their results into 1 to 5 as showed in the below Figure 4.

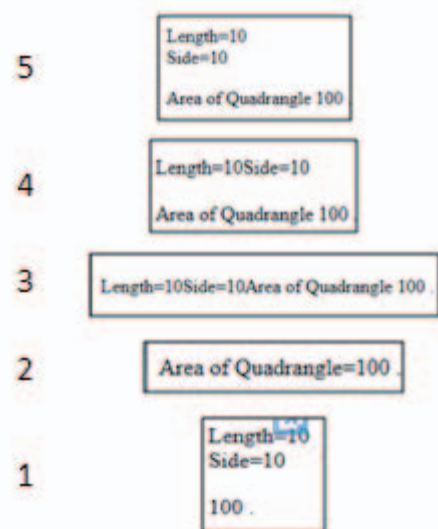


Fig. 4. The Grading of the Results

The definitions of the classification are as follows:

- 5: Completely same as the results example, or almost the same as the specifications.

- 4: Close to the examples presented, but is not easy to see.
- 3: Difficult to see the executed results.
- 2: Only the area of the rectangle is outputted.
- 1: Display is correct, but we found difficulties to understand the results.

Table 5 shows the grading and the number of programs.

TABLE V. GRADING OF RESULTS

Appearance Level	BUA	TDA
5	1	1
4	0	1
3	0	2
2	7	2
1	0	0

Both the BUA group and the TDA group have one student whose program was graded as 5 (BUA 12.5%, TDA 16.7%). For the evaluation 4, the BUA group 0 (0%), and 1 in the TDA group (16.7%). For the evaluation 3, the BUA group has 0 people (0%), while TDA has 2 people (33.3%). For the evaluation 2, the BUA group has 7 (87.5%), and TDA has 2 (33.3%). Nobody was graded as 1 (0%).

This table shows that the TDA group performed better than the BUA group, and the results of the TDA group were closer to the example results.

#### D. Analysis of Problem 2

In this section, we analyze the results of the problem 2. The problem 2 was correct answer by 1 people with both groups .

We show a result of two-people of the source code (figure 5, figure 7) and the practice result (figure 6, figure 8) .

```

<script>
var hen1;
var hen2;
var menseki;
hen1 = 10;
hen2 = 10;
menseki = hen1 * hen2;
document.write("Area of Quadrangle ",menseki);
if(hen1 == hen2)
document.write("Square");
else
document.write("Rectangle");
</script>

```

Fig. 5. Source Code of BUA

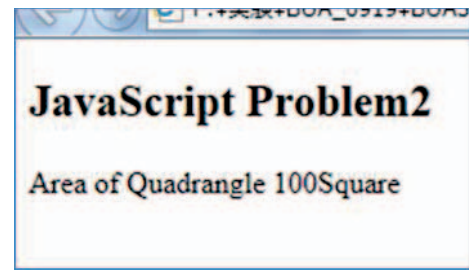


Fig. 6. Execution result of BUA

```

<script>
var hen1;
hen1=10;
document.write("Length: "+hen1+"<br>");

var hen2=20;
document.write("Side: "+hen2+"<br>");

var menseki;
menseki=hen1*hen2;
document.write("Area of a Quadrangle "+ menseki+".<br>");
if(hen1==hen2)
{
document.write("The Square");
}
else
{
document.write("The Rectangle");
}
</script>

```

Fig. 7. Source code of TDA

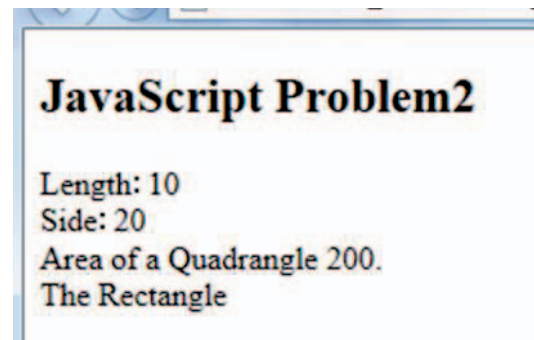


Fig. 8. Execution result of TDA

#### 1) Readability

At first, we compared a source code of BUA with TDA. It is clear that a source code of TDA has high apparently. As for the source code of BUA, there are not indents on empty lines whereas there are indents and empty lines in the source code of TDA.

In addition, we watch the block element of the program. BUA does not have a block. And TDA is divided into a block every each function.

It may be said that there was improvement of the readability by having used TDA.



## 2) If statement

In terms of the "if sentence" TDA has higher readability than BUA.

The description of the "if sentence" of BUA works definitely, but is hard to distinguish the place of the if sentence, because, BUA does not use a curly-brace "{ }". In addition, BUA does not use the indentation. It reduced the readability of "if sentence".

TDA uses curly brace. It brings higher readability of "if sentence". In addition, TDA use the indentation and readability improves.

We taught the TDA students the description method of "if sentence" that has high readability efficiently by sample program at the same time in the learning time of the interval.

## 3) Execution Result

Next, we compared the practice result of TDA with BUA.

TDA showed better readability and the results were simile to the specifications. The readability of the coding of TDA was better because the TDA students imitate the sample program

## E. Discussion of Results and Analysis

The use of the sample program in TDA can say that follows from a result and analysis.

- Readability of source code can improve.
- Reuse it's are close to the specification
- Possible to teach the basic concept of reuse

They unconsciously used the blank line or indentation. They naturally learned by imitating and reusing by the sample program. The learners of the group of TDA can this way learn programming having higher readability and maintainability than the learner of the BUA group.

## F. Summary of Analysis

We analyzed the source code of 41 learners. The source code analysis was performed mainly on the readability. Program readability is high affect the quality and maintainability of the software. Readability was highly improved in the TDA group than the BUA group.

## V. ANALYSIS OF QUESTIONNAIRE

At the end of the experiment, we carried out a questionnaire to know how the learner felt each learning approach.

We analyzed based in the result of the questionnaire.

The questions of the questionnaire are as follows:

Q1. Gender

Q2. Are you interested in programming languages?

Q3. Did you understand the learning method?

Q4. Did you easily understand the sample program?

(Only TDA)

Q5. Do you want to continue to learn in the future?

## Q6. free description

The question 2 is a question of "no" or "yes". "Question 3" from "Question 5" were asked to answer by "1 from 5". The number goes higher when the students feel better (5 is best). The question 4 was made only to TDA group. The question 6 was asked to describe the comments for each approach as a free description.

We showed in Appendix 2 (not including the results of question 6) the results of the questionnaire.

The question 2 was about interest in the programming. In the question 2 answered "yes" 8 persons of BUA group and 4 persons of TDA group.

The answer of the question 3, 4 and 5 answered level of un by BUA group was five people. The Table 6 shows the result of BUA.

TABLE VI. ANSWER QUESTION 3 OF BUA

#	BU7	BU10	BU14	BU20	BU21
<b>Problem 1</b>	Y	N	Y	Y	N
<b>Problem 2</b>	Y	N	N	N	N
<b>Problem 3</b>	N	N	N	N	N

Five students responded "Yes I understand" and 3 at of 5 cleared we Problem1.

In TDA Group 5 people with 4 and 5. The results of the 5 people is showed in the table 7.

TABLE VII. ANSWER QUESTION 3 OF TDA

#	TD3	TD4	TD12	TD16	TD19
<b>Problem 1</b>	N	Y	N	Y	Y
<b>Problem 2</b>	N	Y	N	N	N
<b>Problem 3</b>	N	N	N	N	N

Five people answered 'understood' in TDA group. There out of 5 were able to clear the problem 1.

Question 3 is intended to reveal the differences in understanding level between BUA and TDA. There was not a big difference between BUA and TDA

Question 4 was asked only to TDA was conducted. It was 12 out of 20 people that replied with 4 and 5. Approximately 60% said it was easy to understand.

The results of 12 people who responded yes it was easy understand is shown in Table 8.

TABLE VIII. RELATIONSHIP OF THE QUESTIONNAIRE AND UNDERSTANDING OF THE SAMPLE OF TDA

#	T D 3	T D 4	T D 5	T D 6	T D 7	T D 8	T D 9	T D1 1	T D1 2	T D1 5	T D1 7	T D1 9
Pro ble m 1	N	Y	N	N	Y	Y	N	N	N	N	Y	Y
Pro ble m 2	N	Y	N	N	N	N	N	N	N	N	N	N
Pro ble m 3	N	N	N	N	N	N	N	N	N	N	N	N

Five students out of 12 were able to correctly answer the problem 1. Five students who made a correct program must have truly understood the sample programs, write 7 students have the higher possibilities of thickly that "I think I understand everything."

"I think I understood everything" is one of the serious issues that TDA has.

Question5 is answered by 5 or 4 about the future learning, In the BUA group 8 people out of 21 people and TDA group 4 out of 20 people. BUA, having higher rate in programming that TDA, shows higher rate in future learning.

Finally, there were the following comments on a free description of question 6 in the BUA group.

- I did not understand the format of JavaScript, and could not figure out what I was doing.
- I did not know what would be changed if I changed some parts of the source code.
- It was interesting.
- I was glad when I could complete. I want to try again.

The replies of the TDA group were.

- It was fun because I programed a lot.
- It was not easy for me to understand the sample programs.
- I like deep thinking. I was extremely glad when I correctly completed, and was deeply disappointed when I could not.

There were no significant differences between the TDA group and BUA group.

There was a reply that request wore detain comment listed in the sample program. It may be said that the description form of the comment is a big issue.

The replies to Q6 found that a student who could successfully save the problems, and/or who has an intent in programming shows a positive opinion.

Follow the comment of "I am happy when the program worked". We assumed that we could give success experiences.

#### A. Study through the entire questionnaire

After the experiment, there was not a big difference between the answers BUA group and TDA group. However, it may be said that TDA is applicable in the programming language learning of the beginner because the answers almost similar to BUA, a conventional learning methods from the TDA Group are.

A part of the TDA learners who answered that "I property understand the sample programs" could contently answered the problems.

There was a reply saying that "I was very happy when the program worked" this can be interpreted the success experience using sample programs will highly motivate the program learning and keep the motivation. The success experience and interest in programming are quite important in learning [6] [7]/

#### VI. ISSUES OF TDA

We list issues of TDA:

- What and how should the comments of the sample programs be written.
- How should we handle the "I-think-I-learned-everything" syndrome?
- How the learners should gain the ability of trouble shooting in programming errors?

The important issue is format and content of the program description comment in the sample program. The comment is an element letting understand sample programs. Comments students must briefly explain the program, and easy to understand.

When TDA is applied, same students tend to mistakenly assume that "I think I understand everything" this is perhaps because they were satisfied when they successfully run the sample programs. We must design the sample programs not cause an illusion to learners.

The mistakes such as in the description of the programming language learning are an important element. It is important to have self-solving skills in the practice of programming. Since the sample programs do not have any bugs, it may be necessary to offer sample program with several bugs so that the learners can bring up the ability to solve the problem by themselves

In order to solve the problems as described above, it is necessary to create an optimal sample program. It is important that the comment be described in conjunction with the source code.

#### VII. FUTURE WORKS

We applied our learning process to small-scale programs. We need to apply to large-scale programs.

#### VIII. CONCLUSION

We proposed a learning process based on the top down approach that applies sample programs. To prove its effect, we conducted experiments of 41 high school students.

When we analyzed the results of the experiments, we focused on the readability of the source code. We found that the programs made by the TDA group showed better readability than BUA: this is because the students in the TDA group unintentionally followed the example coding, and learned how to write easy-to-read programs..

#### ACKNOWLEDGMENT

We would like to thank those who participated in the experiment in particular, Mr. Yuji Kitagawa, Mr. Akira Okui, and the students at Tokai-Sugao high school. Also Mr. Masaru Kobayashi, Mr. Hirakata Suma and Ranato Nojiri for their support to provide the experiment.

#### REFERENCE

- [1] Lahtinen, E., Ala-Mutka, K., and Järvinen, "A study of the difficulties of novice programmers", ACM SIGCSE Bulletin, Sept 2005, Volume 37 Issue 3, pp. 14-18
- [2] Reek, Margaret M, "A top-down approach to teaching programming." ACM SIGCSE Bulletin 27.1 (1995), pp. 6-9.
- [3] Saito, Daisuke, and Tsuneo Yamauara, "A New Approach to Programming Language Education for Beginners with Top-Down Learning.", International Journal of Engineering Pedagogy 2013, pp. 16-21
- [4] Zakas, Nicholas C, Maintainable JavaScript, O'Reilly Media, Inc., 2012.
- [5] Stefanov, Stoyan, JavaScript patterns, O'Reilly Media, Inc.", 2010.
- [6] Jenkins, Tony. "The motivation of students of programming." ACM SIGCSE Bulletin. Vol. 33. No. 3. ACM, 2001, pp. 53-56
- [7] Alderman, M. Kay, Motivation for achievement: Possibilities for teaching and learning, Routledge, 2013

#### APPENDIX

APPENDIX 1, PROBLEM RESULT OF BUA

	BU 1	BU 2	BU 3	BU 4	BU 5	BU 6	BU 7	BU 8	BU 9	BU 10	BU 11	BU 12	BU 13	BU 14	BU 15	BU 16	BU 17	BU 18	BU 19	BU 20	BU 21
Problem 1	N	N	N	N	N	N	Y	Y	N	N	Y	N	Y	Y	Y	Y	N	N	N	Y	N
Problem 2	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Problem 3	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N

Y = Yes

N = No

APPENDIX 2, PROBLEM RESULT OF TDA

	TD 1	TD 2	TD 3	TD 4	TD 5	TD 6	TD 7	TD 8	TD 9	TD 10	TD 11	TD 12	TD 13	TD 14	TD 15	TD 16	TD 17	TD 18	TD 19	TD 20
Problem 1	N	N	N	Y	N	N	Y	Y	N	N	N	N	N	N	N	Y	Y	N	Y	N
Problem 2	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Problem 3	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N

Y = Yes

N = No

APPENDIX 3, QUESTIONNAIRE RESULT OF BUA

	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15	B16	B17	B18	B19	B20	B21
Q1	F	F	F	F	F	M	M	M	M	M	M	M	F	F	F	F	F	M	M	M	M
Q2	N	N	Y	Y	N	N	Y	Y	N	Y	N	N	N	N	N	N	N	N	Y	Y	Y
Q3	1	2	3	2	2	3	4	3	1	4	1	2	3	4	1	1	2	2	3	4	4
Q5	2	2	3	5	4	3	4	3	4	4	1	3	3	3	1	3	3	2	4	4	4

F = Female, M = Male

Y = Yes, N = No

APPENDIX 4, QUESTIONNAIRE RESULT OF TDA

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20
<b>Q1</b>	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	M	M	M	M
<b>Q2</b>	N	N	N	Y	N	N	N	Y	N	N	N	N	N	N	N	N	Y	N	Y	N
<b>Q3</b>	1	2	4	5	3	2	3	3	1	3	3	4	1	1	3	4	3	2	4	1
<b>Q4</b>	1	3	4	4	4	4	4	4	4	3	4	4	1	3	4	3	4	3	4	1
<b>Q5</b>	1	2	1	4	3	3	4	4	2	3	3	3	1	1	3	2	3	2	4	2

F = Female, M = Male

Y = Yes, N = No