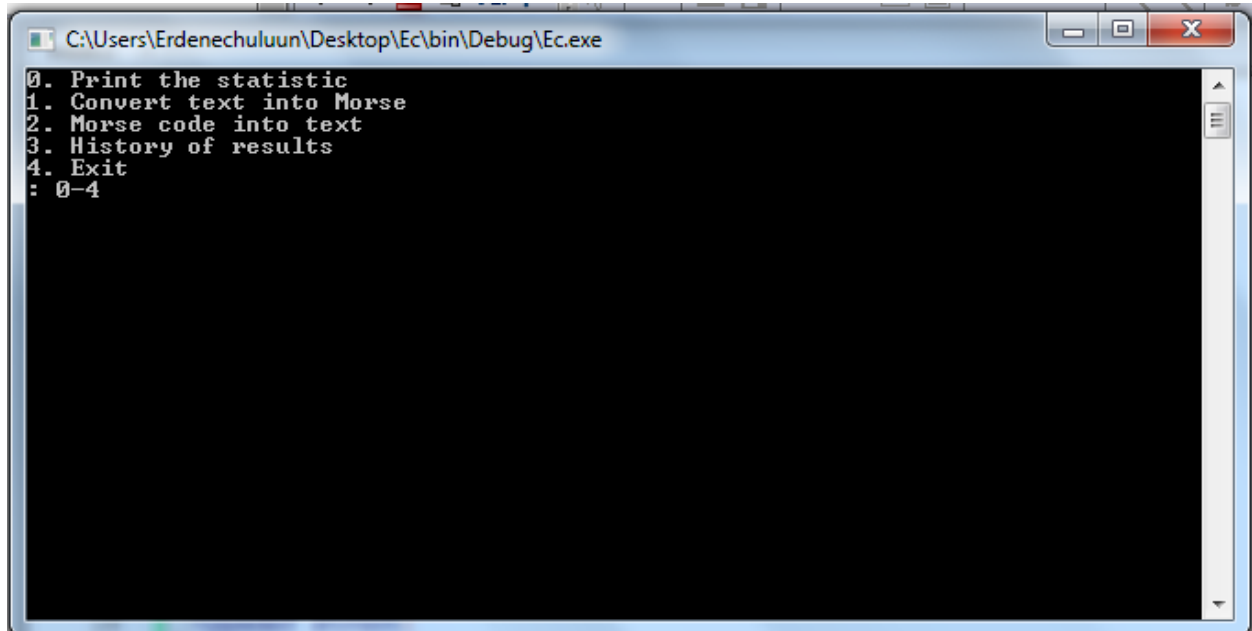


User's manual

Main user interface consists of menu. Menu has 4 options.

Options are to print statistics of Morse code and alphabetical characters. convert text into Morse code, convert Morse code into text, prints the all history of user's inputs and output and exit from the program.



Firstly, user must enter number from 1 to 4 to choose operations.

Operations:

1. Prints the statistics of Morse code and alphabetical characters.
2. Converts text into Morse code
3. Converts Morse code into text
4. Prints the all history of user's inputs and outputs
5. Exits

Secondly, user may enter text or code to convert.

Warning, program runs until user enter option number 4. Other options won't terminate program.

Instructions:

If user chose **menu item 0**, program will print the statistics

Menu item 1, user may enter texts to convert. Program doesn't convert numbers, but only Latin alphabetical letters.

Menu item 2, user may enter Morse code with space to avoid confusion with characters.

For example: 3 dots are “S” character or triple “E” characters.

Menu item 3, program prints the history

Menu item 4, program terminates.

Picture guide to use program:

<pre>0. Print the statistic 1. Convert text into Morse 2. Morse code into text 3. History of results 4. Exit : 1</pre>	<pre>0. Print the statistic 1. Convert text into Morse 2. Morse code into text 3. History of results 4. Exit : 1 Give a text to convert</pre>
--	---

User enter the number, then enter the text to convert.

<pre>0. Print the statistic 1. Convert text into Morse 2. Morse code into text 3. History of results 4. Exit : 1 Give a text to convert Hello World!</pre>	<pre>0. Print the statistic 1. Convert text into Morse 2. Morse code into text 3. History of results 4. Exit : 1 Give a text to convert Hello World! Result:-... .-.. --- .-- --- .-. .-.. -..</pre>
--	---

```
0. Print the statistic
1. Convert text into Morse
2. Morse code into text
3. History of results
4. Exit
: 1
Give a text to convert
Hello World!
Result: .... . .-... .-.. --- .-- --- .-. .-.. -..
0. Print the statistic
1. Convert text into Morse
2. Morse code into text
3. History of results
4. Exit
:
```

Program runs until user enter operation number 4.

<pre>0. Print the statistic 1. Convert text into Morse 2. Morse code into text 3. History of results 4. Exit : 2 Give a Morse code to convert-... .-.. --- .-- --- .-. .-.. -..</pre>	<pre>0. Print the statistic 1. Convert text into Morse 2. Morse code into text 3. History of results 4. Exit : 2 Give a Morse code to convert-... .-.. --- .-- --- .-. .-.. -.. Result: helloworld</pre>
--	---

Example of using operation 2.

```

0. Print the statistic
1. Convert text into Morse
2. Morse code into text
3. History of results
4. Exit
: 3
History begins:Hello World .....-.-. -.-. --- .-- --- -. ....-..
helloworld .....-.-. -.-. --- .-- --- -. ....-..

0. Print the statistic
1. Convert text into Morse
2. Morse code into text
3. History of results
4. Exit
:

```

Operation number 3.

```

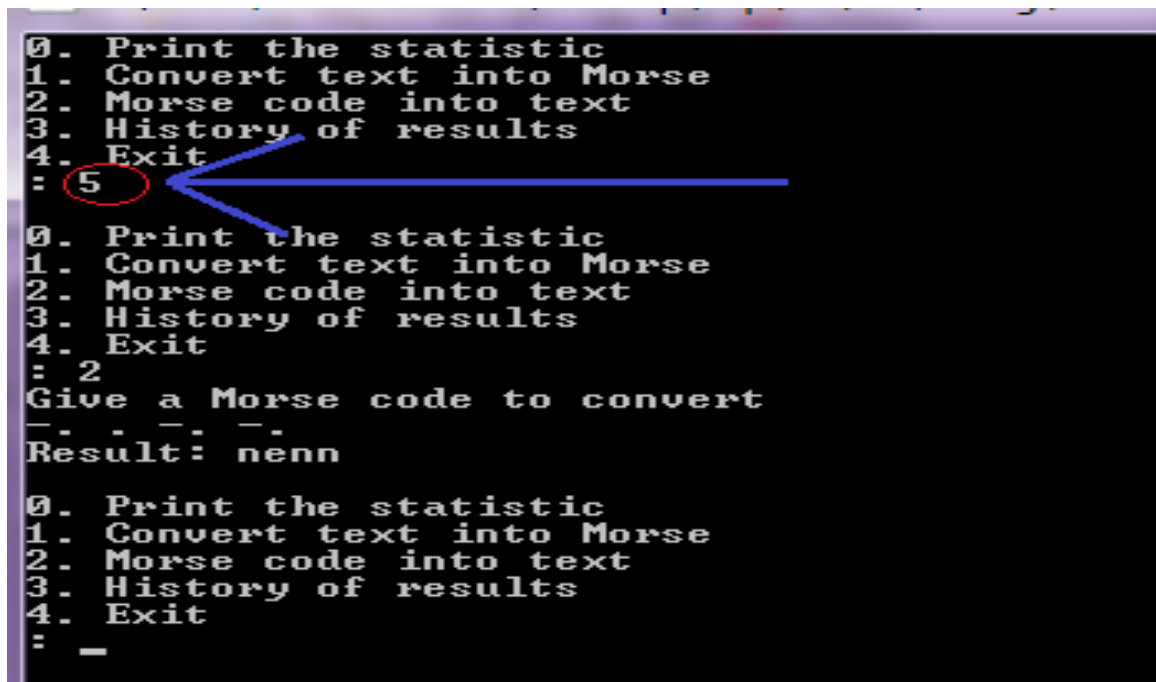
0. Print the statistic
1. Convert text into Morse
2. Morse code into text
3. History of results
4. Exit
: 4

Process returned 0 (0x0)   execution time : 69.047 s
Press any key to continue.

```

Operation number 4, program terminates.

Exceptions and wrong use:



```

0. Print the statistic
1. Convert text into Morse
2. Morse code into text
3. History of results
4. Exit
: 5

0. Print the statistic
1. Convert text into Morse
2. Morse code into text
3. History of results
4. Exit
: 2
Give a Morse code to convert
-.-.-.-
Result: nenn

0. Print the statistic
1. Convert text into Morse
2. Morse code into text
3. History of results
4. Exit
: -

```

If user enter wrong number of operations, program will refresh.