Data 24th of October, 2017 Name: Teofana-Ionela Moisi

# Problem statement:

A math teacher needs a program that will help students test different properties of complex numbers, provided in the a+bi form (assume a, b integers for simplicity). The program manages a list of complex numbers and allows its user to repeatedly execute the following functionalities (each functionality is exemplified):

1. Add numbers to the list.

add <number>  
insert <number> at <position>

e.g.

add 4+2i – adds 4+2i at the end of the list  
insert 1+1i at 1 – insert number 1+i at position 1 in the list; positions are numbered from 0.

2. Modify elements from the list.

remove <position>  
remove <start position> to <end position> replace <old number> with <new number>

e.g.

remove 1 – removes the number at position 1.  
remove 1 to 3 – removes the numbers at positions 1,2, and 3.  
replace 1+3i with 5-3i – replaces all the occurrences of number 1+3i with the number 5-3i.

3. Write numbers having different properties.

list  
list real <start position> to <end position> list modulo [ < | = | > ] <number>

e.g.

list – write the list of numbers.  
list real 1 to 5 – writes the real numbers (imaginary part =0) between positions 1 and 5 in the list.  
list modulo < 10 – writes all numbers having modulo <10 from the list.  
list modulo = 5 – writes all numbers having modulo =10 from the list.

4. Obtain different characteristics of sub-lists.

sum <start position> to <end position> product <start position> to <end position>

e.g.

sum 1 to 5 – writes the sum of the numbers between positions 1 and 5 in the list. product 1 to 5 - writes the product of numbers between position 1 and 5 in the list.

5. Filter the list.

filter real  
filter modulo [ < | = | > ] <number>

e.g.

filter real – keep only real numbers (imaginary part =0) in the list.  
filter modulo < 10 – keep only those numbers having modulo <10 in the list. filter modulo > 6 – keep only those numbers having modulo >6 in the list.

6. Undo the last operation that modified program data.  
undo – the last operation that has modified program data will be reversed. The user has to be able to undo all operations performed since program start by repeatedly calling this function.

# Feature list

|  |
| --- |
| **Feature** |
| F1. Shows a message in order to help the user. |
| F2. **Shows the menu with all the commands.** |
| F3. **Adds a complex number at the end of the list.** |
| F4. Inserts a complex number at the mentioned position. |
| F5. Removes the complex number situated on the metioned position. |
| F6. Removes the numbers from a starting position to an ending position. |
| F7. Replaces an old number with a new one. |
| F8. Lists all the complex numbers. |
| F9. Lists all the real numbers form a starting position to an ending positon. |
| F10.Lists all the numbers having the module >, =, <, <=, >= than a given number. |
| F11.Sums all the numbers from a starting position to an ending position. |
| F12.Clculates the product of all the numbers from a staring position to an ending position. |
| F13. Filters the real numbers from the list. |
| F14.Filters the numbers having the module >, =, <, <=, >= than a given number. |
| F15. Undoes the previous command. |
| F16. Exits the programme. |

## Running scenario 1

|  |  |  |  |
| --- | --- | --- | --- |
|  | **User** | **Program** | **Description** |
| 1 |  | Welcome to the complex number apploication! For further information, please type Help! | When the user opens the programme, it will show a message, in order to help the user to use it. |
| 2 | Help! |  | In order to see the commands, the user will type Help! |
| 3 |  | add <number>  insert <number> at <position>  add <numbers>  remove <position>  remove\_more <positions>  remove <start position> to <end position>  replace <old number> with <new number>  list  list real <start position> to <end position>  list modulo [<|=|>] <number>  sum <start position> to <end position>  product <start position> to <end position>  filter real  filter modulo [<|=|>] <number>  undo  Exit!  Help! | The programme shows all the commands. |
| 4 | add 5+9j |  | Adds the number at the end of the list |
| 5 | list |  | The programme will list all the numbers in the list. |
| 6 |  | 3 + 5 j  -3 + 6 j  6 j  11  4 + 5 j  -23 + 7 j  2 + 10 j  0  -3 j  -8  5.0 + 9.0 j | The list with all the numbers. |
| 7 | replace -3j with 8+9j |  | The programme will replace the old number with the new one. |
| 8 | list | 3 + 5 j  -3 + 6 j  6 j  11  4 + 5 j  -23 + 7 j  2 + 10 j  0  8.0 + 9.0 j  -8  5.0 + 9.0 j | The programme will show the new list. |
| 9 | List real 1 to 10 |  |  |
| 10 |  | 11, 0, -8 | The programme will list all the real numbers. |
| 11 | Sum 3 to 6 |  |  |
| 12 |  | -1+57j | The programme will calculate the sum and display it. |
| 13 | Filter real |  | The programme will delete the numbers in the list which are not real. |
| 14 | list | 11, 0, -8 |  |
| 15 | undo |  | The programme will undo the filter opperation |
| 16 | list | 3 + 5 j  -3 + 6 j  6 j  11  4 + 5 j  -23 + 7 j  2 + 10 j  0  8.0 + 9.0 j  -8  5.0 + 9.0 j |  |

## Tasks

|  |  |
| --- | --- |
| **Id** | **Description** |
| T1 | Adds numbers at the end of the list or inserts numbers at a certain position. |
| T2 | Removes a certain number or more numbers between 2 given positions. |
| T3 | Replaces an old number with a new one. |
| T4 | Displays the whole list, only the real numbers or the numbers which have the module >,<,=<,=>,= than a given number. |
| T5 | Calculates and returns the sum and the product of the numbers between two given positions. |
| T6 | Filters the real of modulo numbers. |
| T7 | Undoes the last operation. |

1. **Test case table for function sum(list, start\_poz, end\_poz)**

|  |  |
| --- | --- |
| **Data: list, start\_poz, end\_poz** | **Result: numbers** |
| [[3, 5], [**-**3, 6], [0, 6], [11, 0], [4, 5], [**-**23, 7], [2, 10], [0, 0], [0, **-**3], [**-**8, 0]]  3, 7 | -6+22j |
| [[3, 5], [**-**3, 6], [0, 6], [11, 0], [4, 5], [**-**23, 7], [2, 10], [0, 0], [0, **-**3], [**-**8, 0]]  5, 8 | -21+14j |
| 3 + 5 j  -3 + 6 j  6 j  4 + 5 j  -23 + 7 j  -8  4.0 + 9.0 j  3, 6 | -23+21j |
| -3 + 6 j  6 j  4 + 5 j  -23 + 7 j  -8  4.0 + 9.0 j  5.0 + 6.0 j  3.0 + 8.0 j  4, 7 | 4+23j |
|  |  |
|  |  |
|  |  |
|  |  |