Prac 07 Design

B, Bukanga

221005009

Problem description

The number of items in the array must be specified from a command-line argument.

2. The program should make use of a menu system. Each menu option can roughly be mapped to the rest of the requirements in this list.

3. The user must have the ability to initiate a process in the program to reset the value of each array item to zero.

4. The user must allow the program to fill an array with random whole numbers in between a lower bound number and upper bound number provided by the user. The programmer must ensure the upper bound random number are within a range decided by the programmer.

5. The user must have an option that will force the program to output a horizontal histogram of the array values.

6. The user must have an option that will force the program to output a vertical histogram of values in the array.

7. The program should run until the user specifically chooses to quit the program.

8. The histogram bars must use three different characters, depending on range of values in the histogram

Bars that indicate values in the bottom third must use a '!!' character.

Bars that indicate values in the middle third must use a '@@' character.

Bars that indicate values in the top third must use a '##' character.

Inputs and outputs

Input	
Upper Bound	Standard input Stream
Lower Bound	Standard input Stream
Output	
N/A	

Case v (Display Vertical)

Input	

“v” or “V”	Standard input stream
Output	
Vertical Histogram graph gets printed	Standard Outputted

Case h(Display Horizontal)

Input	
“h” or “H”	Standard input stream
Output	
Horizontal Histogram graph gets printed	Standard Outputted

Case R (Reset)

Input	
“r” or “R”	Standard input stream
Output	
Resets the array	N/A

Data Format

Identifier	Data type	Description
chInput	Char	Select h, v or r
arrNum	Integer	Number of items in the array
Void Display		Displays the data
Void OutputCharacter		OutPuts Characters depending on the value generated in the array
Void InitialiseHistogram		Initialises the Histogram
GetRand	Integer	Creates a random value in each position in the arrNum between 1 and 0
Void Reset	Integer	Resets the array and regenerates random numbers, to display a new graph

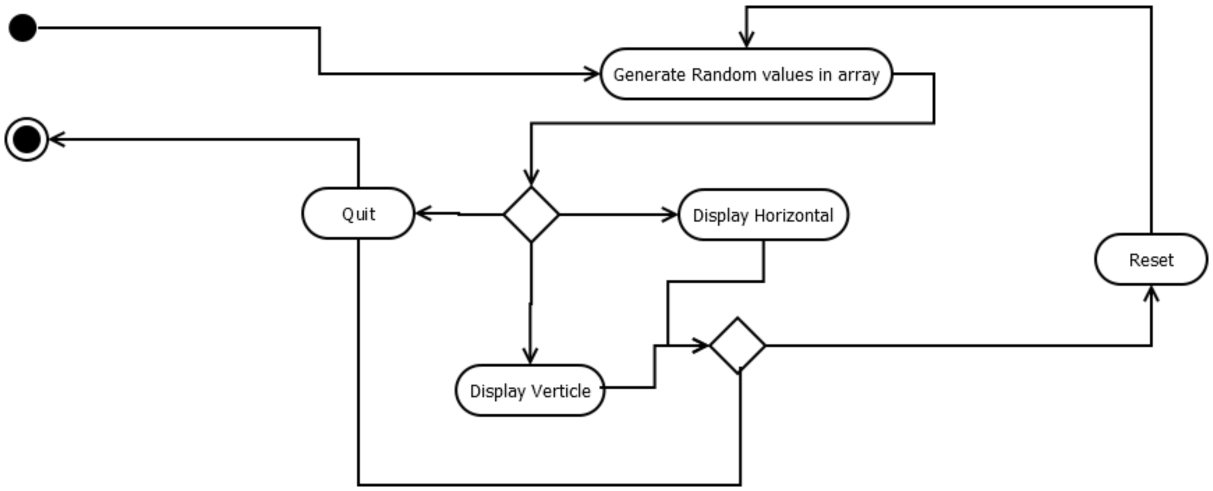
Pseudo Code

```
arrHistogram[] → RandomVal

int range = UpperBound – LowerBound
```

```
        if (RandomVal <= (Range / 3) + lowerbound)
                return Character “!”
        else if (RandomVal >= (Range * (2 / 3) + lowerbound)
                return Character “@”
        else
                return Character “#”
for each item in array
        Display Character << “ ”
```

UML


Generate Random values in array
Quit
Display Horizontal
Reset
Display Verticle