***Testing Documentation***

|  |  |
| --- | --- |
| Test 1 | |
| Input | ./lab2 – no argument was given |
| Output | No argument was passed |
| Screenshots | In the beginning, only ./lab2 was written, this means that only 1 argument was passed which is argument[0], then this is the program name that is printed. |

|  |  |
| --- | --- |
| Test 2 | |
| Input | ./lab2 - a |
| Output | Argument a passed |
| Screenshots | The first argument is the program name. The second argument passed was ‘a’, which gets printed. Along with the ‘ch’ value which is the ASCII code of the input – ‘a’ = 97 |

|  |  |
| --- | --- |
| Test 3 | |
| Input | ./lab2 - A |
| Output | Argument A called |
| Screenshots | The first argument is the program name. The second argument passed was ‘A’, which gets printed. Along with the ‘ch’ value which is the ASCII code of the input – ‘A’ = 65 |

|  |  |
| --- | --- |
| Test 4 | |
| Input | ./lab2 - b |
| Output | Argument b called |
| Screenshots | The first argument is the program name. The second argument passed was ‘b’, which gets printed. Along with the ‘ch’ value which is the ASCII code of the input – ‘b’ = 98 |

|  |  |
| --- | --- |
| Test 5 | |
| Input | ./lab2 - x |
| Output | Argument x called |
| Screenshots | The first argument is the program name. The second argument passed was ‘x’, which gets printed. Along with the ‘ch’ value which is the ASCII code of the input – ‘x’ = 120 |

|  |  |
| --- | --- |
| Test 6 | |
| Input | ./lab2 - v |
| Output | Argument v called |
| Screenshots | The first argument is the program name. The second argument passed was ‘v’, which gets printed. Along with the ‘ch’ value which is the ASCII code of the input – ‘v’ = 118 |

|  |  |
| --- | --- |
| Test 7 | |
| Input | ./lab2 - z |
| Output | Argument z called |
| Screenshots | The first argument is the program name. The second argument passed was ‘z’, which gets printed. Along with the ‘ch’ value which is the ASCII code of the input – ‘z’ = 122 |

|  |  |
| --- | --- |
| Test 8 | |
| Input | ./lab2 - R |
| Output | Argument R called |
| Screenshots | The first argument is the program name. The second argument passed was ‘R’, which gets printed. Along with the ‘ch’ value which is the ASCII code of the input – ‘R’ = 82 |

|  |  |
| --- | --- |
| Test 9 | |
| Input | ./lab2 – h – lower case |
| Output | Not right argument called |
| Screenshots | The first argument is the program name. The second argument passed was ‘h’, which has no validity/method – so it prints the ‘ch’ value, ASCII value =104, then ‘Not right Argument called’. |

|  |  |
| --- | --- |
| Test 10 | |
| Input | ./lab2 – P – upper case |
| Output | Not right argument called |
| Screenshots | The first argument is the program name. The second argument passed was ‘P’, which has no validity/method – so it prints the ‘ch’ value, ASCII value =80, then ‘Not right Argument called’. |

|  |  |
| --- | --- |
| Test 11 | |
| Input | ./lab2 - passing argument in different order so from between – v |
| Output | Argument v called |
| Screenshots | The first argument is the program name. The second argument passed was ‘v’, which gets printed. Along with the ‘ch’ value which is the ASCII code of the input – ‘v’ = 118; This one was to check if the methods still work if the letters are passed not in order |

|  |  |
| --- | --- |
| Test 12 | |
| Input | ./lab2 – - passing argument in different order so from between – b |
| Output | Argument b called |
| Screenshots | The first argument is the program name. The second argument passed was ‘b’, which gets printed. Along with the ‘ch’ value which is the ASCII code of the input – ‘b’ = 98; This one was to check if the methods still work if the letters are passed not in order |

The command prompt command lines called :

