

```

1  //////////////////////////////////////
2  //////////////////////////////////////
3  //                                UTILITIES 121                                //
4  //                                utilities121.cpp                                //
5  //                                //
6  // Written By :                      Environment :                      //
7  // Date .....:                      Compiler ...:                      //
8  //////////////////////////////////////
9  //////////////////////////////////////
10
11 //////////////////////////////////////
12 //                                #includes                                //
13 //////////////////////////////////////
14 #include "utilities121.h"
15 #include <iostream>
16 #include <sstream>
17 #include <stdexcept>
18 #include <limits>
19 #include <random>
20
21 using namespace std;
22 //////////////////////////////////////
23 //                                Function Definitions                                //
24 //////////////////////////////////////
25 /*****
26 *                                void pause(void)                                *
27 *                                *
28 *      Description ....: Delay the system so that the user can read what      *
29 *                        is written to standard output.                        *
30 *                                *
31 *                                *
32 * Written By : Michael R. Nowak      Environment : Mac OS X 10.10.5      *
33 * Date .....: 2017/10/01            Compiler ...: Homebrew GCC 6.3.0_1    *
34 *****/
35 void pause(void)
36 {
37     std::string temp;
38     std::cout << std::endl << "[Press RETURN to continue]" << std::endl;
39     std::getline(std::cin, temp);
40 }
41
42 /*****
43 *                                char upChar(char c)                                *
44 *                                *
45 *      Description ....: Converts an uppercase character to a lower
46 *                        case
47 *                        character. If a non-character is inputted, that input is
48 *                        returned
49 *                        the user
50 *
51 * Written By : Maria Dmitrievskaia      Environment : Windows 10
52 * Date .....: 02/19/19                  Compiler ...: Putty
53 *****/
54 char upChar(char c)
55 {
56     if (c >= 97 && c <= 122)
57         c-=32;
58     return c;
59 }
60
61 /*****
62 *                                char lowChar(char c)                                *
63 *                                *
64 *      Description ....: Converts a lowercase character to an upper
65 *                        case
66 *                        character. If a non-character is inputted, that input is
67 *                        returned
68 *                        the upper
69 *
70 *
71 *
72 *
73 *
74 *
75 *
76 *
77 *
78 *
79 *
80 *
81 *
82 *
83 *
84 *
85 *
86 *
87 *
88 *
89 *
90 *
91 *
92 *
93 *
94 *
95 *
96 *
97 *
98 *
99 *
100 *
101 *
102 *
103 *
104 *
105 *
106 *
107 *
108 *
109 *
110 *
111 *
112 *
113 *
114 *
115 *
116 *
117 *
118 *
119 *
120 *
121 *
122 *
123 *
124 *
125 *
126 *
127 *
128 *
129 *
130 *
131 *
132 *
133 *
134 *
135 *
136 *
137 *
138 *
139 *
140 *
141 *
142 *
143 *
144 *
145 *
146 *
147 *
148 *
149 *
150 *
151 *
152 *
153 *
154 *
155 *
156 *
157 *
158 *
159 *
160 *
161 *
162 *
163 *
164 *
165 *
166 *
167 *
168 *
169 *
170 *
171 *
172 *
173 *
174 *
175 *
176 *
177 *
178 *
179 *
180 *
181 *
182 *
183 *
184 *
185 *
186 *
187 *
188 *
189 *
190 *
191 *
192 *
193 *
194 *
195 *
196 *
197 *
198 *
199 *
200 *
201 *
202 *
203 *
204 *
205 *
206 *
207 *
208 *
209 *
210 *
211 *
212 *
213 *
214 *
215 *
216 *
217 *
218 *
219 *
220 *
221 *
222 *
223 *
224 *
225 *
226 *
227 *
228 *
229 *
230 *
231 *
232 *
233 *
234 *
235 *
236 *
237 *
238 *
239 *
240 *
241 *
242 *
243 *
244 *
245 *
246 *
247 *
248 *
249 *
250 *
251 *
252 *
253 *
254 *
255 *
256 *
257 *
258 *
259 *
260 *
261 *
262 *
263 *
264 *
265 *
266 *
267 *
268 *
269 *
270 *
271 *
272 *
273 *
274 *
275 *
276 *
277 *
278 *
279 *
280 *
281 *
282 *
283 *
284 *
285 *
286 *
287 *
288 *
289 *
290 *
291 *
292 *
293 *
294 *
295 *
296 *
297 *
298 *
299 *
300 *
301 *
302 *
303 *
304 *
305 *
306 *
307 *
308 *
309 *
310 *
311 *
312 *
313 *
314 *
315 *
316 *
317 *
318 *
319 *
320 *
321 *
322 *
323 *
324 *
325 *
326 *
327 *
328 *
329 *
330 *
331 *
332 *
333 *
334 *
335 *
336 *
337 *
338 *
339 *
340 *
341 *
342 *
343 *
344 *
345 *
346 *
347 *
348 *
349 *
350 *
351 *
352 *
353 *
354 *
355 *
356 *
357 *
358 *
359 *
360 *
361 *
362 *
363 *
364 *
365 *
366 *
367 *
368 *
369 *
370 *
371 *
372 *
373 *
374 *
375 *
376 *
377 *
378 *
379 *
380 *
381 *
382 *
383 *
384 *
385 *
386 *
387 *
388 *
389 *
390 *
391 *
392 *
393 *
394 *
395 *
396 *
397 *
398 *
399 *
400 *
401 *
402 *
403 *
404 *
405 *
406 *
407 *
408 *
409 *
410 *
411 *
412 *
413 *
414 *
415 *
416 *
417 *
418 *
419 *
420 *
421 *
422 *
423 *
424 *
425 *
426 *
427 *
428 *
429 *
430 *
431 *
432 *
433 *
434 *
435 *
436 *
437 *
438 *
439 *
440 *
441 *
442 *
443 *
444 *
445 *
446 *
447 *
448 *
449 *
450 *
451 *
452 *
453 *
454 *
455 *
456 *
457 *
458 *
459 *
460 *
461 *
462 *
463 *
464 *
465 *
466 *
467 *
468 *
469 *
470 *
471 *
472 *
473 *
474 *
475 *
476 *
477 *
478 *
479 *
480 *
481 *
482 *
483 *
484 *
485 *
486 *
487 *
488 *
489 *
490 *
491 *
492 *
493 *
494 *
495 *
496 *
497 *
498 *
499 *
500 *
501 *
502 *
503 *
504 *
505 *
506 *
507 *
508 *
509 *
510 *
511 *
512 *
513 *
514 *
515 *
516 *
517 *
518 *
519 *
520 *
521 *
522 *
523 *
524 *
525 *
526 *
527 *
528 *
529 *
530 *
531 *
532 *
533 *
534 *
535 *
536 *
537 *
538 *
539 *
540 *
541 *
542 *
543 *
544 *
545 *
546 *
547 *
548 *
549 *
550 *
551 *
552 *
553 *
554 *
555 *
556 *
557 *
558 *
559 *
560 *
561 *
562 *
563 *
564 *
565 *
566 *
567 *
568 *
569 *
570 *
571 *
572 *
573 *
574 *
575 *
576 *
577 *
578 *
579 *
580 *
581 *
582 *
583 *
584 *
585 *
586 *
587 *
588 *
589 *
590 *
591 *
592 *
593 *
594 *
595 *
596 *
597 *
598 *
599 *
600 *
601 *
602 *
603 *
604 *
605 *
606 *
607 *
608 *
609 *
610 *
611 *
612 *
613 *
614 *
615 *
616 *
617 *
618 *
619 *
620 *
621 *
622 *
623 *
624 *
625 *
626 *
627 *
628 *
629 *
630 *
631 *
632 *
633 *
634 *
635 *
636 *
637 *
638 *
639 *
640 *
641 *
642 *
643 *
644 *
645 *
646 *
647 *
648 *
649 *
650 *
651 *
652 *
653 *
654 *
655 *
656 *
657 *
658 *
659 *
660 *
661 *
662 *
663 *
664 *
665 *
666 *
667 *
668 *
669 *
670 *
671 *
672 *
673 *
674 *
675 *
676 *
677 *
678 *
679 *
680 *
681 *
682 *
683 *
684 *
685 *
686 *
687 *
688 *
689 *
690 *
691 *
692 *
693 *
694 *
695 *
696 *
697 *
698 *
699 *
700 *
701 *
702 *
703 *
704 *
705 *
706 *
707 *
708 *
709 *
710 *
711 *
712 *
713 *
714 *
715 *
716 *
717 *
718 *
719 *
720 *
721 *
722 *
723 *
724 *
725 *
726 *
727 *
728 *
729 *
730 *
731 *
732 *
733 *
734 *
735 *
736 *
737 *
738 *
739 *
740 *
741 *
742 *
743 *
744 *
745 *
746 *
747 *
748 *
749 *
750 *
751 *
752 *
753 *
754 *
755 *
756 *
757 *
758 *
759 *
760 *
761 *
762 *
763 *
764 *
765 *
766 *
767 *
768 *
769 *
770 *
771 *
772 *
773 *
774 *
775 *
776 *
777 *
778 *
779 *
780 *
781 *
782 *
783 *
784 *
785 *
786 *
787 *
788 *
789 *
790 *
791 *
792 *
793 *
794 *
795 *
796 *
797 *
798 *
799 *
800 *
801 *
802 *
803 *
804 *
805 *
806 *
807 *
808 *
809 *
810 *
811 *
812 *
813 *
814 *
815 *
816 *
817 *
818 *
819 *
820 *
821 *
822 *
823 *
824 *
825 *
826 *
827 *
828 *
829 *
830 *
831 *
832 *
833 *
834 *
835 *
836 *
837 *
838 *
839 *
840 *
841 *
842 *
843 *
844 *
845 *
846 *
847 *
848 *
849 *
850 *
851 *
852 *
853 *
854 *
855 *
856 *
857 *
858 *
859 *
860 *
861 *
862 *
863 *
864 *
865 *
866 *
867 *
868 *
869 *
870 *
871 *
872 *
873 *
874 *
875 *
876 *
877 *
878 *
879 *
880 *
881 *
882 *
883 *
884 *
885 *
886 *
887 *
888 *
889 *
890 *
891 *
892 *
893 *
894 *
895 *
896 *
897 *
898 *
899 *
900 *
901 *
902 *
903 *
904 *
905 *
906 *
907 *
908 *
909 *
910 *
911 *
912 *
913 *
914 *
915 *
916 *
917 *
918 *
919 *
920 *
921 *
922 *
923 *
924 *
925 *
926 *
927 *
928 *
929 *
930 *
931 *
932 *
933 *
934 *
935 *
936 *
937 *
938 *
939 *
940 *
941 *
942 *
943 *
944 *
945 *
946 *
947 *
948 *
949 *
950 *
951 *
952 *
953 *
954 *
955 *
956 *
957 *
958 *
959 *
960 *
961 *
962 *
963 *
964 *
965 *
966 *
967 *
968 *
969 *
970 *
971 *
972 *
973 *
974 *
975 *
976 *
977 *
978 *
979 *
980 *
981 *
982 *
983 *
984 *
985 *
986 *
987 *
988 *
989 *
990 *
991 *
992 *
993 *
994 *
995 *
996 *
997 *
998 *
999 *
1000 *

```

```

66 * Written By : Maria Dmitrievskaia      Environment : Windows 10      *
67 * Date .....: 02/19/19                  Compiler ...: Putty           *
68 *****/
69 char lowChar(char c)
70 {
71     if (c >=65 && c <=90)
72         c+=32;
73
74     return c;
75 }
76
77 /*****
78 *          std::string upperStr(std::string s)          *
79 *          *
80 *          Description ....: Converts a string of characters to a string
81 *          containing only uppercase
82 *          characters.          *
83 *          *
84 * Written By : Maria Dmitrievskaia      Environment : Windows 10      *
85 * Date .....: 02/19/02                  Compiler ...: Putty           *
86 *****/
87 string upperStr(string const& Input)
88 {
89     string upper = "";
90     for(int i = 0; i < Input.size(); ++i){
91         upper += upChar(Input[i]);
92     }
93     return upper;
94 }
95
96 /*****
97 *          std::string lowerStr(std::string s)          *
98 *          *
99 *          Description ....: Converts a string of characters to a
100 *          string          *
101 *          containing only lowercase
102 *          characters.          *
103 *          *
104 * Written By : Maria Dmitrievskaia      Environment : Windows 10      *
105 * Date .....: 02/19/19                  Compiler ...: Putty           *
106 *****/
107 string lowerStr(string const& Input)
108 {
109     string lower = "";
110     for(int i = 0; i < Input.size(); ++i){
111         lower += lowChar(Input[i]);
112     }
113     return lower;
114 }
115
116 /*****
117 *          std::string to_str(int i)          *
118 *          *
119 *          Description ....: The function creates a string containing a passed
120 *          integer value.          *
121 *          *
122 * Written By : Wesley Myers              Environment : Windows 10      *
123 * Date .....: 02/19/19                  Compiler ...: Putty           *
124 *****/
125 string toStr(int i)
126 {
127     stringstream ss;
128     ss << i;
129     string s = ss.str();
130     return s;
131 }
132
133 /*****

```

```

132 *          std::string to_str(double d) *
133 * *
134 *      Description ....: The function creates a string containing a passed *
135 *      floating point value. *
136 * *
137 * *
138 * Written By : Wesley Myers Environment : Windows 10 *
139 * Date .....: 02/19/19 Compiler ...: Putty *
140 *****/
141 string toStr(double i)
142 {
143     stringstream ss;
144     ss << i;
145     string s = ss.str();
146     return s;
147 }
148
149 /*****
150 *      int getInt(int low, int high, std::string const& prompt) *
151 * *
152 *      Description ....: Asks user to input a number between passed high and *
153 *      low values. *
154 * *
155 * *
156 * Written By : Wesley and Maria Environment : Windows 10 *
157 * Date .....: 02/19/19 Compiler ...: Putty *
158 *****/
159 int getInt(int low, int high, std::string const& prompt)
160 {
161     cout << prompt;
162     cout << "Enter an integer value between " << low << " and " << high << endl;
163     int input;
164     do
165     {
166         cin >> input;
167         if (cin.bad())
168         {
169             throw runtime_error("stream corrupted");
170             break;
171         }
172         if (cin.fail())
173         {
174
175             cout << "Input you have provided is invalid..." << endl << "Please try
again:" << endl;
176             cin.clear();
177             cin.ignore(numeric_limits<streamsize>::max ( ) , '\n');
178             cin >> input;
179         }
180         if (cin.eof()) {
181             throw runtime_error("unexpected eof");
182             break;
183         }
184
185         if (input <= low || input >= high)
186         {
187             cout << "Input not in range...\nplease try again" << endl;
188         }
189     } while (!cin.good() || (input <= low || input >= high));
190
191     return input;
192 }
193
194
195 /*****
196 *      double getDouble(double low, double high, double tolerance, *
197 *      std::string const& prompt) *
198 * *

```

```

199 *      Description .....:
200 *
201 *
202 * Written By :  Maria and Wesley      Environment :  Windows 10
203 * Date .....:  02/19/19              Compiler ...:  Putty
204 *****/
205 double getDouble(double low, double high, double tolerance, std::string const& prompt)
206 {
207     cout << prompt;
208     cout << "Enter a double value between " << low << " and " << high << endl;
209     double input;
210     double lower_low = low-tolerance;
211     double higher_high = high + tolerance;
212     do
213     {
214         cin >> input;
215         if (cin.bad())
216         {
217             throw runtime_error("stream corrupted");
218             break;
219         }
220         if (cin.fail())
221         {
222             cout << "Input you have provided is invalid..." << endl << "Please try
223             again:" << endl;
224             cin.clear();
225             cin.ignore(numeric_limits<streamsize>::max ( ) , '\n ');
226             cin >> input;
227         }
228         if (cin.eof()) {
229             throw runtime_error("unexpected eof");
230             break;
231         }
232         if ((input <= (lower_low) )|| (input >= (higher_high)))
233         {
234             cout << "Input not in range...\nplease try again" << endl;
235         }
236     } while (!cin.good() || ((input <= lower_low) || (input >= higher_high)));
237
238     return input;
239 }
240
241
242
243 *****/
244 *      void clearScreen(void)
245 *
246 *      Description .....: This function clears the contents of the terminal
247 *      window
248 *
249 * Written By :  Maria Dmitrievskaia   Environment :  Window 10
250 * Date .....:  02/19/19              Compiler ...:  Putty
251 *****/
252 void clearScreen(void)
253 {
254     system("clear");
255 }

```