

# HW\_02

izd3

Use only commands & functions that are shown in the indicated chapter or prior chapters.

## Problem #01 - Chapter 06 Exercise #01 ( Use the stringr package)

```
# Show your work here  
library('stringr')
```

```
## Warning: package 'stringr' was built under R version 4.2.3
```

```
str_c("Letter:",CharacterData004)
```

```
## [1] "Letter:A" "Letter:B" "Letter:C" "Letter:D" "Letter:E" "Letter:F"  
## [7] "Letter:G" "Letter:H" "Letter:I" "Letter:J" "Letter:K" "Letter:L"  
## [13] "Letter:M" "Letter:N" "Letter:O" "Letter:P" "Letter:Q" "Letter:R"  
## [19] "Letter:S" "Letter:T" "Letter:U" "Letter:V" "Letter:W" "Letter:X"  
## [25] "Letter:Y" "Letter:Z"
```

## Problem #02 - Chapter 06 Exercise #04 ( Use the stringr package)

*# Show your work here*

```
str_dup(string = "Ho!", times = c(9,8,7,6,5,4,3,2,1))
```

```
## [1] "Ho!Ho!Ho!Ho!Ho!Ho!Ho!Ho!Ho!" "Ho!Ho!Ho!Ho!Ho!Ho!Ho!Ho!"
## [3] "Ho!Ho!Ho!Ho!Ho!Ho!Ho!"      "Ho!Ho!Ho!Ho!Ho!Ho!"
## [5] "Ho!Ho!Ho!Ho!Ho!"            "Ho!Ho!Ho!Ho!"
## [7] "Ho!Ho!Ho!"                  "Ho!Ho!"
## [9] "Ho!"
```

### Problem #03 - Chapter 06 Exercise #05 ( Use the stringr package)

*# Show your work here*

```
str_c("I like the letter",CharacterData004,".")
```

```
## [1] "I like the letterA." "I like the letterB." "I like the letterC."
## [4] "I like the letterD." "I like the letterE." "I like the letterF."
## [7] "I like the letterG." "I like the letterH." "I like the letterI."
## [10] "I like the letterJ." "I like the letterK." "I like the letterL."
## [13] "I like the letterM." "I like the letterN." "I like the letterO."
## [16] "I like the letterP." "I like the letterQ." "I like the letterR."
## [19] "I like the letterS." "I like the letterT." "I like the letterU."
## [22] "I like the letterV." "I like the letterW." "I like the letterX."
## [25] "I like the letterY." "I like the letterZ."
```

## Problem #04 - Chapter 07 Exercise #04

*# Show your work here*

```
c( T, T, F, F, TRUE ) & c(T,T,T,T,T)
```

```
## [1] TRUE TRUE FALSE FALSE TRUE
```

```
!c(F, T, T,F,T,T) & c( F, T, F, T, T, F )
```

```
## [1] FALSE FALSE FALSE TRUE FALSE FALSE
```

```
c( T, !FALSE, T, TRUE, T, T ) | c(F, F, !T,F,F,!T )
```

```
## [1] TRUE TRUE TRUE TRUE TRUE TRUE
```

```
!c( FALSE, !FALSE, T, TRUE, T, T, FALSE ) | !c( !F, !F, !FALSE, T,!F,!F,!FALSE )
```

```
## [1] TRUE FALSE FALSE FALSE FALSE FALSE TRUE
```

## Problem #05 - Chapter 07 Exercise #05

*# Show your work here*

```
any(c( T, T, F, F, TRUE ) & c(T,T,T,T,T))
```

```
## [1] TRUE
```

```
all(c( T, T, F, F, TRUE ) & c(T,T,T,T,T))
```

```
## [1] FALSE
```

```
any(!c(F, T, T,F,T,T) & c( F, T, F, T, T, F ))
```

```
## [1] TRUE
```

```
all(!c(F, T, T,F,T,T) & c( F, T, F, T, T, F ))
```

```
## [1] FALSE
```

```
any(c( T, !FALSE, T, TRUE, T, T ) | c(F, F, !T,F,F,!T ))
```

```
## [1] TRUE
```

```
all(c( T, !FALSE, T, TRUE, T, T ) | c(F, F, !T,F,F,!T ))
```

```
## [1] TRUE
```

```
any(!c( FALSE, !FALSE, T, TRUE, T, T, FALSE ) | !c( !F, !F, !FALSE, T,!F,!F,!FALSE ))
```

```
## [1] TRUE
```

```
all(!c( FALSE, !FALSE, T, TRUE, T, T, FALSE ) | !c( !F, !F, !FALSE, T,!F,!F,!FALSE ))
```

```
## [1] FALSE
```

## Problem #06 - Chapter 08 Exercise #01

```
# Show your work here  
3==3L
```

```
## [1] TRUE
```

```
c(1, 2, 3, 4, 5, 6)<4
```

```
## [1] TRUE TRUE TRUE FALSE FALSE FALSE
```

```
set_one<-c(1:25)  
set_two<-sqrt(set_one)^2  
set_one >= set_two
```

```
## [1] TRUE FALSE TRUE TRUE FALSE TRUE FALSE FALSE TRUE FALSE TRUE TRUE  
## [13] TRUE TRUE FALSE TRUE TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE  
## [25] TRUE
```

```
set_three<-sqrt(set_one^2)  
set_one<=set_three
```

```
## [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [16] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

## Problem #07 - Chapter 09 Exercise #01

*# Show your work here*

```
typeof(c(1, 2L, 3L, 4L, 5L, 6L, 7L, 8L))
```

```
## [1] "double"
```

```
typeof(c(T, T, T, T, "F"))
```

```
## [1] "character"
```

```
typeof(c(T, 1))
```

```
## [1] "double"
```

```
typeof(c('a', T, 10))
```

```
## [1] "character"
```



Problem #08 - Chapter 09 Exercise #03 (Do not display the 1000 element results.)

```
# Show your work here
```

```
#a
first<-seq(from=1,to=1000, by=1)
#b.
second<-sqrt(first)
#c.
third<-seq(from=2,to=1000,by=2)
third
```

```
##      [1]      2      4      6      8     10     12     14     16     18     20     22     24     26     28     30
##     [16]     32     34     36     38     40     42     44     46     48     50     52     54     56     58     60
##     [31]     62     64     66     68     70     72     74     76     78     80     82     84     86     88     90
##     [46]     92     94     96     98    100    102    104    106    108    110    112    114    116    118    120
##     [61]    122    124    126    128    130    132    134    136    138    140    142    144    146    148    150
##     [76]    152    154    156    158    160    162    164    166    168    170    172    174    176    178    180
##     [91]    182    184    186    188    190    192    194    196    198    200    202    204    206    208    210
##    [106]    212    214    216    218    220    222    224    226    228    230    232    234    236    238    240
##    [121]    242    244    246    248    250    252    254    256    258    260    262    264    266    268    270
##    [136]    272    274    276    278    280    282    284    286    288    290    292    294    296    298    300
##    [151]    302    304    306    308    310    312    314    316    318    320    322    324    326    328    330
##    [166]    332    334    336    338    340    342    344    346    348    350    352    354    356    358    360
##    [181]    362    364    366    368    370    372    374    376    378    380    382    384    386    388    390
##    [196]    392    394    396    398    400    402    404    406    408    410    412    414    416    418    420
##    [211]    422    424    426    428    430    432    434    436    438    440    442    444    446    448    450
##    [226]    452    454    456    458    460    462    464    466    468    470    472    474    476    478    480
##    [241]    482    484    486    488    490    492    494    496    498    500    502    504    506    508    510
##    [256]    512    514    516    518    520    522    524    526    528    530    532    534    536    538    540
##    [271]    542    544    546    548    550    552    554    556    558    560    562    564    566    568    570
##    [286]    572    574    576    578    580    582    584    586    588    590    592    594    596    598    600
##    [301]    602    604    606    608    610    612    614    616    618    620    622    624    626    628    630
##    [316]    632    634    636    638    640    642    644    646    648    650    652    654    656    658    660
##    [331]    662    664    666    668    670    672    674    676    678    680    682    684    686    688    690
##    [346]    692    694    696    698    700    702    704    706    708    710    712    714    716    718    720
##    [361]    722    724    726    728    730    732    734    736    738    740    742    744    746    748    750
##    [376]    752    754    756    758    760    762    764    766    768    770    772    774    776    778    780
##    [391]    782    784    786    788    790    792    794    796    798    800    802    804    806    808    810
##    [406]    812    814    816    818    820    822    824    826    828    830    832    834    836    838    840
##    [421]    842    844    846    848    850    852    854    856    858    860    862    864    866    868    870
##    [436]    872    874    876    878    880    882    884    886    888    890    892    894    896    898    900
##    [451]    902    904    906    908    910    912    914    916    918    920    922    924    926    928    930
##    [466]    932    934    936    938    940    942    944    946    948    950    952    954    956    958    960
##    [481]    962    964    966    968    970    972    974    976    978    980    982    984    986    988    990
##    [496]    992    994    996    998  1000
```

```
#d.
fourth<-seq(from=1,to=999,by=2)
fourth
```

```
##      [1]      1      3      5      7      9     11     13     15     17     19     21     23     25     27     29     31     33     35
```

```
## [19] 37 39 41 43 45 47 49 51 53 55 57 59 61 63 65 67 69 71
## [37] 73 75 77 79 81 83 85 87 89 91 93 95 97 99 101 103 105 107
## [55] 109 111 113 115 117 119 121 123 125 127 129 131 133 135 137 139 141 143
## [73] 145 147 149 151 153 155 157 159 161 163 165 167 169 171 173 175 177 179
## [91] 181 183 185 187 189 191 193 195 197 199 201 203 205 207 209 211 213 215
## [109] 217 219 221 223 225 227 229 231 233 235 237 239 241 243 245 247 249 251
## [127] 253 255 257 259 261 263 265 267 269 271 273 275 277 279 281 283 285 287
## [145] 289 291 293 295 297 299 301 303 305 307 309 311 313 315 317 319 321 323
## [163] 325 327 329 331 333 335 337 339 341 343 345 347 349 351 353 355 357 359
## [181] 361 363 365 367 369 371 373 375 377 379 381 383 385 387 389 391 393 395
## [199] 397 399 401 403 405 407 409 411 413 415 417 419 421 423 425 427 429 431
## [217] 433 435 437 439 441 443 445 447 449 451 453 455 457 459 461 463 465 467
## [235] 469 471 473 475 477 479 481 483 485 487 489 491 493 495 497 499 501 503
## [253] 505 507 509 511 513 515 517 519 521 523 525 527 529 531 533 535 537 539
## [271] 541 543 545 547 549 551 553 555 557 559 561 563 565 567 569 571 573 575
## [289] 577 579 581 583 585 587 589 591 593 595 597 599 601 603 605 607 609 611
## [307] 613 615 617 619 621 623 625 627 629 631 633 635 637 639 641 643 645 647
## [325] 649 651 653 655 657 659 661 663 665 667 669 671 673 675 677 679 681 683
## [343] 685 687 689 691 693 695 697 699 701 703 705 707 709 711 713 715 717 719
## [361] 721 723 725 727 729 731 733 735 737 739 741 743 745 747 749 751 753 755
## [379] 757 759 761 763 765 767 769 771 773 775 777 779 781 783 785 787 789 791
## [397] 793 795 797 799 801 803 805 807 809 811 813 815 817 819 821 823 825 827
## [415] 829 831 833 835 837 839 841 843 845 847 849 851 853 855 857 859 861 863
## [433] 865 867 869 871 873 875 877 879 881 883 885 887 889 891 893 895 897 899
## [451] 901 903 905 907 909 911 913 915 917 919 921 923 925 927 929 931 933 935
## [469] 937 939 941 943 945 947 949 951 953 955 957 959 961 963 965 967 969 971
## [487] 973 975 977 979 981 983 985 987 989 991 993 995 997 999
```

```
#e.
fifth<-seq(from=10,to=-50,by=-1)
fifth
```

```
## [1] 10 9 8 7 6 5 4 3 2 1 0 -1 -2 -3 -4 -5 -6 -7 -8
## [20] -9 -10 -11 -12 -13 -14 -15 -16 -17 -18 -19 -20 -21 -22 -23 -24 -25 -26 -27
## [39] -28 -29 -30 -31 -32 -33 -34 -35 -36 -37 -38 -39 -40 -41 -42 -43 -44 -45 -46
## [58] -47 -48 -49 -50
```

## Problem #09 - Chapter 09 Exercise #05

*# Show your work here*

*#a)*

```
seq(from=-0.123, to=68, length.out=9)
```

```
## [1] -0.123000  8.392375 16.907750 25.423125 33.938500 42.453875 50.969250
## [8] 59.484625 68.000000
```

*#b)*

```
seq(from=90,by=-12,length.out=19)
```

```
## [1]  90  78  66  54  42  30  18   6  -6 -18 -30 -42 -54 -66 -78
## [16] -90 -102 -114 -126
```

*#c)*

```
seq(from=9,by=3,to=21.2)
```

```
## [1]  9 12 15 18 21
```

*#d)*

```
vec_1<-seq(from=10,to=2001,by=10)
sqrt(vec_1)
```

```
## [1]  3.162278  4.472136  5.477226  6.324555  7.071068  7.745967  8.366600
## [8]  8.944272  9.486833 10.000000 10.488088 10.954451 11.401754 11.832160
## [15] 12.247449 12.649111 13.038405 13.416408 13.784049 14.142136 14.491377
## [22] 14.832397 15.165751 15.491933 15.811388 16.124515 16.431677 16.733201
## [29] 17.029386 17.320508 17.606817 17.888544 18.165902 18.439089 18.708287
## [36] 18.973666 19.235384 19.493589 19.748418 20.000000 20.248457 20.493902
## [43] 20.736441 20.976177 21.213203 21.447611 21.679483 21.908902 22.135944
## [50] 22.360680 22.583180 22.803509 23.021729 23.237900 23.452079 23.664319
## [57] 23.874673 24.083189 24.289916 24.494897 24.698178 24.899799 25.099801
## [64] 25.298221 25.495098 25.690465 25.884358 26.076810 26.267851 26.457513
## [71] 26.645825 26.832816 27.018512 27.202941 27.386128 27.568098 27.748874
## [78] 27.928480 28.106939 28.284271 28.460499 28.635642 28.809721 28.982753
## [85] 29.154759 29.325757 29.495762 29.664794 29.832868 30.000000 30.166206
## [92] 30.331502 30.495901 30.659419 30.822070 30.983867 31.144823 31.304952
## [99] 31.464265 31.622777 31.780497 31.937439 32.093613 32.249031 32.403703
## [106] 32.557641 32.710854 32.863353 33.015148 33.166248 33.316662 33.466401
## [113] 33.615473 33.763886 33.911650 34.058773 34.205263 34.351128 34.496377
## [120] 34.641016 34.785054 34.928498 35.071356 35.213634 35.355339 35.496479
## [127] 35.637059 35.777088 35.916570 36.055513 36.193922 36.331804 36.469165
## [134] 36.606010 36.742346 36.878178 37.013511 37.148351 37.282704 37.416574
## [141] 37.549967 37.682887 37.815341 37.947332 38.078866 38.209946 38.340579
## [148] 38.470768 38.600518 38.729833 38.858718 38.987177 39.115214 39.242834
## [155] 39.370039 39.496835 39.623226 39.749214 39.874804 40.000000 40.124805
## [162] 40.249224 40.373258 40.496913 40.620192 40.743098 40.865633 40.987803
## [169] 41.109610 41.231056 41.352146 41.472883 41.593269 41.713307 41.833001
## [176] 41.952354 42.071368 42.190046 42.308392 42.426407 42.544095 42.661458
```

```
## [183] 42.778499 42.895221 43.011626 43.127717 43.243497 43.358967 43.474130
## [190] 43.588989 43.703547 43.817805 43.931765 44.045431 44.158804 44.271887
## [197] 44.384682 44.497191 44.609416 44.721360
```

```
#e)
vec_2<-seq(from=82,to=2001,by=82)
sqrt(vec_2)+25
```

```
## [1] 34.05539 37.80625 40.68439 43.11077 45.24846 47.18107 48.95830 50.61250
## [9] 52.16616 53.63564 55.03331 56.36877 57.64966 58.88215 60.07136 61.22154
## [17] 62.33631 63.41875 64.47151 65.49691 66.49699 67.47352 68.42810 69.36215
```

## Problem #10 - Chapter 09 Exercise #06abc

```
# Show your work here
#a)
c(VectorsOne,VectorsInts)
```

```
## [1] 1 4 5 6 7
```

```
#b)
rep(c(VectorsA,VectorsGirl),length.out=100)
```

```
## [1] "a" "girl" "a" "girl" "a" "girl" "a" "girl"
## [11] "a" "girl" "a" "girl" "a" "girl" "a" "girl"
## [21] "a" "girl" "a" "girl" "a" "girl" "a" "girl"
## [31] "a" "girl" "a" "girl" "a" "girl" "a" "girl"
## [41] "a" "girl" "a" "girl" "a" "girl" "a" "girl"
## [51] "a" "girl" "a" "girl" "a" "girl" "a" "girl"
## [61] "a" "girl" "a" "girl" "a" "girl" "a" "girl"
## [71] "a" "girl" "a" "girl" "a" "girl" "a" "girl"
## [81] "a" "girl" "a" "girl" "a" "girl" "a" "girl"
## [91] "a" "girl" "a" "girl" "a" "girl" "a" "girl"
```

```
#c)
c(VectorsA,VectorsOne,VectorsA,VectorsGirl,VectorsInts,VectorsA)
```

```
## [1] "a" "1" "a" "girl" "4" "5" "6" "7" "a"
```

## Problem #11 - Chapter 09 Exercise #07a

```
# Show your work here  
ans<-seq(from=156, to=10000, by=pi)
```