Project #4 – disk usage

Course	INFO-1156 Object-Oriented Programming in C++
Professor	Janice Manning and Lynn Koudsi
Assigned	Thursday, July 22, 2021
Due	Monday, August 9, 2021
Weight	10%
Student Name	

Project Description

Create a C++ 17[1] console[2] application that shows disk usage (based on the UNIX command). du (abbreviated from disk usage) is inspired by a standard Unix program used to estimate file space usage on a file system. Your application will parse files from a given directory on a hard drive.

```
C:\du>du.exe c:\Temp
0       c:\Temp\Empty
305     c:\Temp\gamma
3821    c:\Temp\HelloWorldC++
5598    c:\Temp\{2.1} HiLo
24      c:\Temp\{8.2} string_demo
14938   c:\Temp
```

Purpose:

- To display allocation space (not absolute file space) for each file and directory contained in a given pathname

Usage:

- du takes at least a single argument, specifying a pathname
- If no pathname is provided, the current directory is used

du reports:

- Pathnames and the sum of the clusters allocated for each directory
- The sum of clusters for the given root directory

Switches:

- Switches provide input options for additional functionality
  - "-" is used before a single letter switch or a collection of single letter switches
  - "--" is used before a word switch

Program Interfaces

```
du [-skhznrb] [--help] [--version] [--block-size=dddd] [folder]*
```

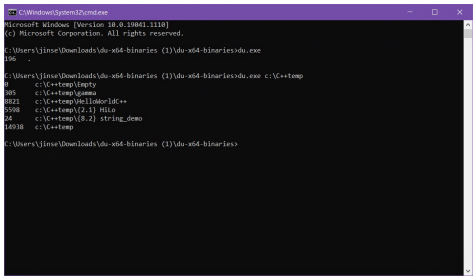
Where:

-s	Display only the final summary of each root folder
-k	Set the cluster size to 1024
-h	Make human readable. Output is in KiB, MiB, GiB, TiB[3] rounded to either one decimal place if less than 10 or zero decimal places otherwise. A zero value has neither a decimal place or a unit. (see demo).
-z	Sort rows by size
-n	Sort rows by folder name
-r	Display rows in reverse order
-b	Output in bytes, not clusters
--help	Display the help listing
--version	Display the version number only in the form d.d.d
--block-size=dddd	Set the cluster size to the specified value > 0.
[folder]*	If no folder is provided the current working directory is used. Otherwise, the specified folder is where the scan begins.

[1] Must be compiled with /std:c++17
[2] Windows platform
[3] KiB is different from a KB. KiB = 1024 or 2^{10} whereas a KB = 1000. MiB = 2^{20}, GiB=2^{30}, and TiB=2^{40}.

Input Rules:

- Cannot use -b and -h switches together
- -k and –block-size are incompatible
- -z and -n switches are incompatible
- No switch may be duplicated
- There cannot be unknown switches



Grading Criteria

Requirements	Weight	Points	Awarded	Grade
Test Cases				
#0: no args	10%	1	1	10%
#1: version switch	2%	1	1	2%
#2: help switch	3%	1	1	3%
#3: default folder: no files/subfolders	6%	1	1	6%
#4: default folder: files/no subfolders	6%	1	1	6%
#5: specified folder: no subfolders	5%	1	1	5%
#6: multiple folders: with subfolders	2%	1	1	2%
#7: -s: no folder	3%	1	1	3%
#8: -s: one folder	5%	1	1	5%
#9: -s: multiple folders	2%	1	1	2%
#10: -k kilobyte switch	5%	1	1	5%
#11: -b: byte switch	5%	1	1	5%
#12: -h: human readable switch	8%	1	1	8%
#13: -h: human readable switch - in folder	1%	1	1	1%
#14: -h: human readable switch - multiple folders	1%	1	1	1%
#15: -z: sort by size	3%	1	1	3%
#16: -n: sort by name	3%	1	1	3%
#17: -r: reverse	2%	1	1	2%
#18: -rz: reverse size	2%	1	1	2%
#19: -rn: reverse name	1%	1	1	1%
#20: --blocksize=512	3%	1	1	3%
#21: --blocksize=2048	2%	1	1	2%
#22: -a: bad switch	2%	1	1	2%
#23: --block-size=x: bad block-size switch	2%	1	1	2%
#24: no -b and -h switches	2%	1	1	2%
#25: no -k and --block-size switches	2%	1	1	2%
#26: no -z and -n switches	1%	1	1	1%
Non-functional requirements				
Multi-file solution	2%	1	1	2%
minimum of 3 spaces between columns	1%	1	1	1%

Columns are aligned	2%	1	1	2%
numeric column width is dynamic	4%	1	1	4%
Human readable uses a precision of 1 if value is less than 10.0, precision of zero otherwise	2%	1	1	2%
Penalties				
Penalties from *C & C++ Grading Guide* ***v2.2.0***	**-5%**	1	0	0%
Executable named other than 'du.exe'	**-10%**	1	0	0%
Late submission:	**-10%**	1	0	0%
Total				**100%**

Difficulties
Moderate
Harder
Hardest

Submission Requirements

1. Submit **entire Visual Studio project directory** to Fanshawe Online
    a. Delete ***all*** debug and release directories.[i]
    b. Submit in a .ZIP, .7z archive file.

[i] Alternatively, you can 'clean' your project for submission by downloading 'vsclean' a Visual Studio Solution Cleaner from www.gats.ca .