

```
import hashlib

print("CSV file validation using MD5 checksums\n\n")

#all csv files were uploaded to this colab notebook to be compared using md5 checksum
csv_files = [
    'histogram_sequential.csv',
    'histogram_pthreads_2.csv',
    'histogram_pthreads_4.csv',
    'histogram_pthreads_8.csv',
    'histogram_openmp_2.csv',
    'histogram_openmp_4.csv',
    'histogram_openmp_8.csv',
    'histogram_mpi_2.csv',
    'histogram_mpi_4.csv',
    'histogram_mpi_8.csv',
    'histogram_cuda_128.csv',
    'histogram_cuda_256.csv',
    'histogram_cuda_512.csv',
    'histogram_cuda_1024.csv',
    'histogram_cuda_shared.csv',
    'histogram_cuda_tiled.csv',
]
#this function computes the md5 checksum of a file
def compute_md5(filename):
    md5_hash = hashlib.md5()
    try:
        with open(filename, "rb") as f:
            for chunk in iter(lambda: f.read(4096), b ""):
                md5_hash.update(chunk)
        return md5_hash.hexdigest()
    except FileNotFoundError:
        return None

checksums = {}
#fill the dictionary with the checksums to check later
for filename in csv_files:
    checksum = compute_md5(filename)
    if checksum:
        checksums[filename] = checksum
        print(f"{filename}<35} {checksum}")
    else:
        print(f"{filename}<35} FILE NOT FOUND")

print("\n\nValidation results\n")
valid=True
checksum_list = list(checksums.values())
c = checksum_list[0]
for i in range(1, len(checksum_list)):
    if checksum_list[i] != c:
        valid = False
        break
if valid:
    print("CSV Histograms are all identical\nAll files have the same checksum")
else:
    print("CSV Histograms are not identical\nFiles have different checksums")
```

CSV file validation using MD5 checksums

histogram_sequential.csv	1ded4b48fd25b6089ff689a207e89b50
histogram_pthreads_2.csv	1ded4b48fd25b6089ff689a207e89b50
histogram_pthreads_4.csv	1ded4b48fd25b6089ff689a207e89b50
histogram_pthreads_8.csv	1ded4b48fd25b6089ff689a207e89b50
histogram_openmp_2.csv	1ded4b48fd25b6089ff689a207e89b50
histogram_openmp_4.csv	1ded4b48fd25b6089ff689a207e89b50
histogram_openmp_8.csv	1ded4b48fd25b6089ff689a207e89b50
histogram_mpi_2.csv	1ded4b48fd25b6089ff689a207e89b50
histogram_mpi_4.csv	1ded4b48fd25b6089ff689a207e89b50
histogram_mpi_8.csv	1ded4b48fd25b6089ff689a207e89b50
histogram_cuda_128.csv	1ded4b48fd25b6089ff689a207e89b50
histogram_cuda_256.csv	1ded4b48fd25b6089ff689a207e89b50
histogram_cuda_512.csv	1ded4b48fd25b6089ff689a207e89b50
histogram_cuda_1024.csv	1ded4b48fd25b6089ff689a207e89b50
histogram_cuda_shared.csv	1ded4b48fd25b6089ff689a207e89b50
histogram_cuda_tiled.csv	1ded4b48fd25b6089ff689a207e89b50

Validation results

CSV Histograms are all identical
All files have the same checksum

Start coding or generate with AI.