

Programming assignment - The Image Storage Estimator

Your assignment is to create a piece of software for estimating image storage size. The tool should handle three types of images, base line jpeg images, jpeg 2000 images and uncompressed bitmaps. It should also take special measures to handle stacks of images.

The following are the requirements:

- The tool should take its input from the standard input and write any output to standard output.
- The program starts with printing a header with your name and a short description of the software. The user should be able to enter one or many rows containing information about the images for which to estimate storage size. A line takes the form of "ImageType Width Height", with each part separated by space. The user can signal end of input by entering "q", at which point the total storage size of the images is printed.
- Valid input for image type is "J" or "JPG" for baseline jpeg, "JP2" or "JPEG2000" for jpeg 2000, and "BMP" for uncompressed bitmaps. Width and height should be any positive integer.
- The tool should also be able to group images in stacks by allowing the user to enter "G" at the start of the line, followed by indices of the images to group. Image indices start with 1 for the first image, two for the second and so on. An image should not be allowed to belong to multiple groups and a group of images cannot be part of another group.
- The storage size for baseline jpeg images is calculated with the formula "width * height * 0.2", truncated to whole bytes.
- The storage size for jpeg 2000 images is calculated with the formula "width * height * 0.4 / ln(ln(width * height + 16))", truncated to whole bytes.
- The storage size for uncompressed bitmaps is calculated by assuming one byte per pixel.
- When images are grouped in a stack, they can be compressed even further with the total size calculated with the following formula "(total storage size of images) / ln(number of base level images in stack + 3)", truncated to whole bytes.

Note that for baseline jpeg images and for uncompressed bitmaps (but not for jpeg 2000), pyramid level images should be accounted for, having the same compression as the base level image. Pyramid level images are a quarter of the size of the larger level, down to a minimum size of 128 pixels in the smallest dimension. So for an image of size 2000x1000, the pyramid levels would be 1000x500 and 500x250, but no more since the next level would be too small (250x125, where 125 is smaller than the minimum allowed size of 128). Division results are always truncated. Here is an example run with input and matching output:

Storage calculator by John Doe. Enter one line for each image/group on the format "type width height", or "G i, i, ...". Exit with "Q".

```
J 2048 1024
JPG 5000 3000
G 1 2
JP2 1000 2000
BMP 2000 1000
```

Total size: 5 753 091 bytes

The object of this task is to test your skills. This includes, among other things, object orientation, design abilities, knowledge of platform, language, and foremost maintainability of the code.

As you will be judged based on your submission, make sure to produce code having the same quality that we can expect from you in your daily work.

If you feel that something is unclear in the specification, feel free to make assumptions about the problem, preferably based on what you think would be reasonable given the things you already know. However, make sure that you document these design decisions, preferably in a text file together with your submission.