

A solução em Assembly deve consistir da função histograma, que verifica se o limite de tamanho foi respeitado, zera o vetor histograma para então calculá-lo, e também da parte em C++, responsável por chamar a função em Assembly e exibir os resultados. Como solução opcional pode ser criada uma implementação da função histograma em C++, uma função comparativa dos dois histogramas e outra que será responsável por realizar cada processo para cada imagem.

Detalhadamente, a rotina em Assembly do histograma deve ter como entrada quatro parâmetros: número de linhas e de colunas de half-word, e vetores da imagem de um byte e do histograma de uma half-word, onde serão guardados os resultados. Os dois primeiros parâmetros devem ser passados por valor e dois últimos por referência, ou ponteiro ao início dos vetores, sendo que todos passados pelos registradores R0-R3, respectivamente. Assim, deverá ser feita a multiplicação de linhas por colunas para saber o tamanho, e ser armazenado em R0, ou então R0 assumirá zero caso o tamanho seja maior que 65535 bytes. O próximo passo é zerar o histograma, simplesmente escrevendo 0 na memória e respeitando o tamanho de 256 posições de half-words para no próximo passo ler os valores da imagem e acessar cada elemento do histograma com um offset, com o valor do nível de cinza do pixel lido, a partir do primeiro elemento do histograma e acrescentar mais um a seu valor (uma leitura e escrita).

Em C++ não é necessário a criação de classes, por ser um programa pequeno e não ter a necessidade para tal. Portanto serão implementadas somente as quatro funções aqui ditas e a main.

A main deve ficar responsável pela alocação de memória para os dois histogramas que serão gerados, um em C++ e o outro em Assembly. Além disso deve chamar uma outra função que deve ser responsável em gerar, comparar e exibir os resultados para cada imagem, ou seja, chamada com as constantes de número de linhas, de colunas e ponteiro para o início da imagem, e as variáveis de histograma calculado em C++ e outra em Assembly. Tal deve então chamar a rotina em Assembly e a função em C++ que deve ser seu equivalente, uma função de comparação de arrays, passando seu tamanho e os dois histogramas calculados, para então verificar elemento a elemento sua igualdade e retornar códigos referentes a desigualdade ou igualdade. Caso desigual, ou a função em C++ ou em Assembly devem estar má implementadas. Uma última função que deve ser chamada é a de exibir um array, no caso histograma, recebendo o array e seu tamanho, exibindo índice e o valor correspondente a tal pelo cout e possivelmente agrupando índices de valores iguais.

### Observações

Na função de exibição do histograma, não foi possível a utilização de “std::cout” por um erro cuja solução não foi identificada quando era tentado exibir números inteiros, em qualquer formato, apesar de que com strings não há erro algum. Oportunamente é possível a utilização de “printf” em C++, que no caso foi utilizado.

Também foi alterada a primeira imagem contida no arquivo “images.c”.