BCC202 – Estruturas de Dados I (2020-01)

Departamento de Computação - Universidade Federal de Ouro Preto - MG
Professor: **Pedro Silva** (www.decom.ufop.br/)
Estagiário docente: **Geovani Martins** (geovani.martins@aluno.ufop.edu.br)

Trabalho Prático 2 (TP3) - 10 pontos, peso 1.

- Data de entrega: 16/06/2022 até 23:55. O que vale é o horário do run.codes, e não do *seu*, ou do *meu* relógio!!!

- Clareza, identação e comentários no código também vão valer pontos. Por isso, escolha cuidadosamente o nome das variáveis e torne o código o mais legível possível.

- O padrão de entrada e saída deve ser respeitado exatamente como determinado no enunciado. Parte da correção é automática, não respeitar as instruções enunciadas pode acarretar em perda de pontos.

- Durante a correção, os programas serão submetidos a vários casos de testes, com características variadas.

- A avaliação considerará o tempo de execução e o percentual de respostas corretas.

- Eventualmente serão realizadas entrevistas sobre os estudos dirigidos para complementar a avaliação;

- O trabalho é em grupo de até 3 (três) pessoas.

- Entregar um relatório.

- Os códigos fonte serão submetidos a uma ferramenta de detecção de plágios em software.

- Códigos cuja autoria não seja do aluno, com alto nível de similaridade em relação a outros trabalhos, ou que não puder ser explicado, acarretará na perda da nota.

- Códigos ou funções prontas específicas de algoritmos para solução dos problemas elencados não são aceitos

- Não serão considerados algoritmos parcialmente implementados.

- Procedimento para a entrega:.

1. Submissão: via **run.codes**.
2. Os nomes dos arquivos e das funções devem ser especificados considerando boas práticas de programação.
3. Funções auxiliares, complementares aquelas definidas, podem ser especificadas e implementadas, se necessário.
4. A solução deve ser devidamente modularizada e separar a especificação da implementação em arquivos *.h* e *.c* sempre que cabível.
5. Os arquivos a serem entregues, incluindo aquele que contém *main()*, devem ser compactados (*.zip*), sendo o arquivo resultante submetido via **run.codes**.
6. Você deve submeter os arquivos *.h*, *.c* e o *.pdf* (relatório) na raiz do arquivo *.zip*. Use os nomes dos arquivos *.h* e *.c* exatamente como pedido.
7. Caracteres como acento, cedilha e afins não devem ser utilizados para especificar nomes de arquivos ou comentários no código.

- Bom trabalho!

Busca Binária para o Projeto de Arte Sacra

Uma equipe de artesões planeja reproduzir algumas peças de arte sacra do importante artista mineiro Antônio Francisco Lisboa, mais conhecido como Aleijadinho, que tem notáveis obras em pedra sabão, entalhes em madeira, altares e igrejas (ver Figura 1). Assim como Aleijadinho, os artesões planejam utilizar a pedra sabão como material de trabalho. A plasticidade dessa rocha, refletida na baixa dureza, permite que ela seja facilmente riscada e moldada, garantindo sua aplicação na confecção de esculturas.

Figura 1: Os 12 Profetas em pedra sabão de tamanho quase natural, obra de Aleijadinho em Congonhas, Minas Gerais.

Durante a elaboração do projeto, os artesões identificaram um importante problema de logística e produção: os blocos de pedra sabão a serem talhados necessitam ter o mesmo volume e obedecer às dimensões do local de exposição. As câmaras de exposição devem apresentar o formato retangular definido na Figura 2. Suas medidas estão em função de um fator x centímetros, sendo o comprimento $a = 1x$, a largura $b = 2x$ e a altura $c = 3x$. Nesse caso, o bloco de pedra sabão deve ter comprimento e largura exatamente iguais as da câmara, porém a altura **h** deve mudar conforme a matéria prima disponível. Note que a altura **h** não pode ultrapassar a altura da câmara.

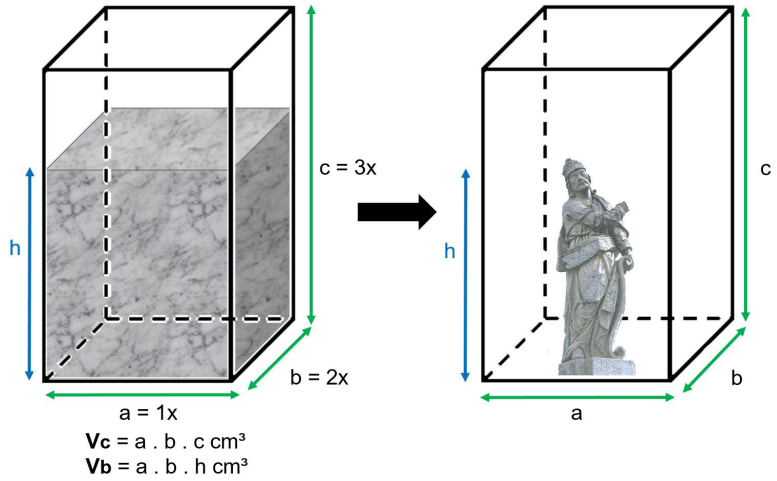

Figura 2: Cálculo do volume e altura de um bloco de pedra sabão. Legenda: `Vc` = Volume da câmara; `Vb` = volume do bloco de pedra sabão; `a` = comprimento da câmara; `b` = largura da câmara; `c` = altura da câmara; `h` = altura do bloco de pedra sabão).

Uma vez que os artesões têm disponível um volume de **V** cm^3 de pedra sabão, essa matéria prima deverá ser dividida igualmente em **n** blocos retangulares para a produção das esculturas de arte sacra. Assim sendo, a sua tarefa é auxiliar a equipe de artesões na identificação da altura de corte **h** ideal para um dado volume **V** de pedra sabão disponível.

Imposições e comentários gerais

Neste trabalho, as seguintes regras devem ser seguidas:

- Seu programa não pode ter *memory leaks*, ou seja, toda memória alocada pelo seu código deve ser

corretamente liberada antes do final da execução. (Dica: utilize a ferramenta *valgrind* para se certificar de que seu código libera toda a memória alocada)

- Um grande número de *Warnings* ocasionará a redução na nota final.

O que deve ser entregue

- Código fonte do programa em C (bem identado e comentado).
- A documentação do trabalho (relatório - **formato:** PDF) deve conter:
  - **Implementação:** descrição sobre a implementação do programa. Não faça "print screens" de telas. Ao contrário, procure resumir ao máximo a documentação, fazendo referência ao que julgar mais relevante. É importante, no entanto, que seja descrito o funcionamento das principais funções e procedimentos utilizados, bem como decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado. Muito importante: os códigos utilizados na implementação devem ser inseridos na documentação.
  - **Impressões gerais**: descreva o seu processo de implementação deste trabalho. Aponte coisas que gostou bem como aquelas que o desagradou. Avalie o que o motivou, conhecimentos que adquiriu, entre outros.
  - **Análise**: deve ser feita uma análise dos resultados obtidos com este trabalho.
  - **Conclusão**: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.

Como deve ser feita a entrega

Verifique se seu programa compila e executa na linha de comando antes de efetuar a entrega. Quando o resultado for correto, entregue via *run.codes* até a 16/06/2022 até 23:55 um arquivo **.ZIP**. Esse arquivo deve conter: (i) os arquivos *.c* e *.h* utilizados na implementação, e (ii) o relatório em **PDF**.

Detalhes da implementação

O código-fonte deve ser modularizado corretamente em três arquivos: *principal.c*, *busca.h* e *busca.c*. O arquivo *principal.c* deve apenas invocar as funções e procedimentos definidos no arquivo *busca.h*. A separação das operações em funções e procedimentos está a cargo do aluno, porém, não deve haver acúmulo de operações dentro de uma mesma função/procedimento.

As informações sobre os blocos de pedra sabão devem ser armazenadas em um vetor alocado dinamicamente (e posteriormente desalocado) para cada caso de teste.

O limite de tempo para solução de cada caso de teste é de apenas **um segundo**. Utilize suas habilidades de programação e de análise de algoritmos para desenvolver um algoritmo correto e rápido!

Entrada

A primeira linha de entrada contém um inteiro C que determina a quantidade de casos de teste. Em seguida, cada linha de caso de teste deve conter as seguintes informações: **n**, **V** e **x** que representa, respectivamente, o número de blocos, o volume disponível de matéria prima em cm^3 e o fator em cm para o cálculo do comprimento das arestas e/ou do volume da câmara.

Entrada no Terminal

```
4
12 989736 36.9
5 900000 40.716
3 88206 15
27 40132.7 0
```

Saída

Para cada caso de teste, exiba uma única linha com a altura h em cm, de modo que, a equipe de artesões consiga atender as especificações necessárias para a exposição das obras de arte sacra. Faça arredondamento dos resultados e imprima com exatamente 3 casas decimais após o ponto.

Exiba "Altura do corte igual ou maior que a câmara de exposição" sempre que a altura de corte for igual ou ultrapassar os limites da câmara de exposição. Nesse caso, os artesões considerarão o bloco de pedra sabão com o mesmo volume da câmara de exposição.

Exiba "Impossível definir a altura do corte" para casos impossíveis de serem tratados com os dados fornecidos.

Saída esperada no Terminal

```
30.287
54.289
Altura do corte igual ou maior que a camara de exposicao
Impossivel definir a altura do corte
```

Diretivas de Compilação

```
$ gcc -c busca.c -Wall
$ gcc -c principal.c -Wall
$ gcc busca.o principal.o -o exe
```