

Documentação

Igor Guilherme Bianchi 558400

Exercício C

Para a resolução deste exercício foi utilizado um algoritmo guloso. Define-se uma struct COMIDA, com as variáveis relativas a comida, volume e valor, e uma variável custoBeni, que é a divisão do valor pelo volume. Primeiramente os dados de cada entrada são passados para vetores. No que se refere à comida, os dados são passados para um vetor de struct, para a parte de alunos são criados dois vetores float, vEstomago (volume do estômago de cada aluno) e valorTotal (valor máximo alcançado por cada aluno, resultado do problema).

Passados os valores para os vetores, o vetor comida é ordenado pelo método quicksort em relação ao custoBeni. Como o vetor agora está ordenado de forma crescente e a solução quer o maior valor possível, é plausível que o procedimento de repetição seja de forma decrescente.

```
for(j=0; j<qtdAlunos; j++){  
    for(i = qtdComida-1; i >= 0; i--){
```

Se o volume da comida é menor que o volume do estômago, então o valor da comida já entra para o resultado e volume do estômago é reduzido para o restante do que ainda não foi ocupado.

```
if(comida[i].volume <= vEstomago[j]){  
    valorTotal[j] += comida[i].valor;  
    vEstomago[j] -= comida[i].volume;  
}
```

Caso contrário, o valor é calculado pelo custoBeni * vEstomago, que representa o valor por cm³ da comida multiplicando o que resta de volume para ser preenchido. É chamado o break para não haver mais verificações, já que o vetor está ordenado e o maior valor já foi atingido.

```
else{  
    valorTotal[j] += vEstomago[j] * comida[i].custoBeni;  
    break;
```

}

O teste foi realizado abrindo o arquivo de entrada dentro do código, apresentou 100% de coerência com a saída esperada.