

Esta prova tem o objetivo de avaliar a capacidade do aluno de desenvolver uma solução elegante e que leve em conta todos os assuntos visto na disciplina. Particularmente, assuntos como o uso eficiente de funções, criação de estruturas, alocação dinâmica de memória, uso de diretivas de compilação, modularização, makefile e documentação.

A forma de entrega é a mesma utilizada no trabalho prática, isto é, os códigos deverão ser enviados ao professor em pacote. O pacote deve ser compactado com tar(1) e bzip2(1) em um arquivo chamado login1.tar.bz2.

O critério de avaliação consiste na legibilidade, elegância, eficiência, modularidade, coesão e acoplamento entre módulos, e uso adequado de estruturas de dados serão levados em conta na avaliação. Os algoritmos de manipulação das estruturas de dados devem ser tão eficientes quanto possível, usando todos os conceitos vistos nas disciplinas de Algoritmos do Curso. Ponteiros e alocação dinâmica de memória devem ser usados onde for eficiente e conveniente. Estruturas dinâmicas abordadas em Alg II, devidamente utilizadas, colaboram fortemente para uma avaliação positiva. Os valores da prova são (i) Funcionamento correto do programa: 30 pontos (ii) Qualidade do código: 60 pontos (iii) Qualidade da documentação: 10 pontos

### **Questão:**

Uma empresa de telemarketing utiliza uma lista de possíveis compradores para os produtos vendidos e seus respectivos telefones. A empresa também possui um time de N vendedores para ligar para esse conjunto de pessoas. Para cada ligação a um comprador da lista sabe-se o tempo T em minutos que ela vai durar. Os vendedores são identificados por números de 1 a N e fazem as ligações da seguinte forma:

- Inicialmente, todos os vendedores estão inativos.
- Sempre que um vendedor realizar uma ligação, ele ficará ocupado pelos T minutos descritos na lista para aquela ligação. O tempo entre duas ligações consecutivas do mesmo vendedor é desprezível.
- Um vendedor não pode fazer mais de uma ligação ao mesmo tempo.
- Um vendedor que esteja inativo deverá fazer a ligação que estiver no topo da lista. Caso mais de um vendedor esteja inativo no mesmo instante, o vendedor com o menor identificador dentre os vendedores inativos deverá fazer a ligação que estiver no topo da lista.
- Assim que uma ligação é atribuída a um vendedor, ela é removida da lista.
- Um vendedor fica inativo sempre que termina uma ligação.

Por exemplo, suponha que um time de 4 vendedores deve fazer 6 ligações, cujos tempos sejam 5, 2, 3, 3, 4, 9. Como inicialmente nenhum vendedor está ocupado, o primeiro vendedor fará a ligação de 5 minutos, o segundo vendedor a ligação de 2 minutos e os vendedores de número 3 e 4 farão ligações de 3 minutos. Como o segundo vendedor terminará a sua ligação antes dos demais, ele fará a quinta ligação, de 4 minutos e, por fim, o terceiro vendedor (cujo tempo é igual ao do quarto vendedor, mas o número é menor) fará a sexta ligação, de 9 minutos.

Escreva um programa que, dados o número de vendedores, o número de ligações e a duração de cada ligação, determine o número de ligações feitas por cada vendedor.

A entrada contém um único conjunto de dados, que deve ser lido do dispositivo de

entrada padrão. A primeira linha da entrada contém dois inteiros,  $N$  e  $L$  indicando o número de vendedores e o número de ligações a serem realizadas ( $1 \leq N \leq 1.000$ ,  $1 \leq L \leq 1.000.000$ ). As  $L$  linhas seguintes contém um inteiro  $T$  cada ( $1 \leq T \leq 30$ ), em que  $T$  representa a duração de cada ligação.

Seu programa deve imprimir, na saída padrão,  $N$  linhas, uma para cada vendedor, contendo dois inteiros  $I$  e  $P$  representando o número do vendedor e o número de ligações realizadas por este vendedor. Os vendedores devem ser apresentados em ordem crescente de identificador, começando a partir de 1.