|  |  |
| --- | --- |
| Logotipo, nome da empresa  Descrição gerada automaticamente | Campus de Cascavel  Centro de Ciências Exatas e Tecnológicas - CCET  Curso de Ciência da Computação  Disciplina: Estruturas de Dados  Professor: Josué Castro |

**Trabalho Prático de Laboratório 2**

Resolva os problemas abaixo em linguagem Go (Golang).

**Questão 1 (1 pt.):**

A disseminação dos computadores se deve principalmente à capacidade de eles se comportarem como outras máquinas, vindo a substituir muitas destas. Esta flexibilidade é possível porque podemos alterar a funcionalidade de um computador, de modo que ele opere da forma que desejarmos: essa é a base do que chamamos programação.

**Tarefa:**

Sua tarefa é escrever um programa que faça com que o computador opere como uma calculadora simples. O seu programa deve ler expressões aritméticas e produzir como saída o valor dessas expressões, como uma calculadora faria. O programa deve implementar apenas um subconjunto reduzido das operações disponíveis em uma calculadora: somas e subtrações.

**Entrada de Dados:**

A entrada é composta de vários conjuntos de testes, gravados no arquivo “**calcula.in**”. A primeira linha de um conjunto de testes contém um número inteiro *m* (1 ≤ *m* ≤ 100), indicando o número de operandos da expressão a ser avaliada. A segunda linha de um conjunto de testes contém a expressão aritmética a ser avaliada, no seguinte formato:

X1 s1 X2 s2 ... Xm-1 sm-1 Xm onde

* *Xi* , 1 ≤ *i* ≤ *m*, é um *operando* (0 ≤ *Xi* ≤ 100);
* *sj*, 1 ≤ *j* < *m*, é um *operador*, representado pelos símbolos ‘+’ ou ‘–’;
* não há espaços em branco entre operandos e operadores.

O final da entrada é indicado pelo valor *m*=0.

**Saída de Dados:**

Para cada conjunto de testes da entrada seu programa deve produzir três linhas. A primeira linha deve conter um identificador da expressão, no formato “Teste n”, onde n é numerado a partir de 1. Na segunda linha deve aparecer o resultado encontrado pelo seu programa. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente. A saída deve ser gravada no arquivo “**calcula.out**”

**Exemplo:**

|  |  |
| --- | --- |
| **Entrada (calcula.in)** | **Saída (calcula.out)** |
| 3  3+7-22  3  5-10-77  10  1+2+3+4+5+6+7+8+9+10  0 | Teste 1  -12  Teste 2  -82  Teste 3  55 |

**Restrições:**

1 ≤ m ≤ 100

0 ≤ Xi ≤ 100 para todo 1 ≤ i ≤ m

**Questão 2 (1 pt.):**

Sem as proteções da atmosfera e do cinturão magnético que existem na Terra, a Lua fica exposta ao ataque do Sol, que é um astro em constante explosão atômica. As explosões do Sol emitem ondas letais de partículas. Uma pessoa que ficasse desprotegida na superfície da Lua, num lugar onde o Sol incidisse diretamente, sofreria um bombardeio radioativo tão intenso quanto se estivesse nas imediações da usina russa de Chernobyl no momento do acidente que matou 31 pessoas, em 1986. Além da radiação solar, outro efeito desta falta de proteção contra o Sol que existe na Lua é a enorme variação de temperatura. Nas regiões próximas do equador lunar, a variação de temperatura é brutal, passando de cerca de 130 graus positivos durante o dia a 129 graus negativos à noite.

Para estudar com mais precisão as variações de temperatura na superfície da Lua, a NASA enviou à Lua uma sonda com um sensor que mede a temperatura de 1 em 1 minuto. Um dado importante que os pesquisadores desejam descobrir é como se comporta a média da temperatura, considerada em intervalos de uma dada duração (uma hora, meia hora, oito horas, etc.). Por exemplo, para a sequência de medições 8, 20, 30, 50, 40, 20, -10, e intervalos de quatro minutos, as médias são respectivamente 108/4=27, 140/4=35, 140/4=35 e 100/4=25.

**Tarefa:**

Você foi recentemente contratado pela NASA, e sua primeira tarefa é escrever um programa que, conhecidos a sequência de temperaturas medidas pelo sensor, e o tamanho do intervalo desejado, informe qual a maior e qual a menor temperatura média observadas, considerando o tamanho do intervalo dado.

**Entrada de dados:**

A entrada é composta de vários conjuntos de teste, gravados no arquivo “**lua.in**”. A primeira linha de um conjunto de teste contém dois números inteiros positivos *N* e *M*, que indicam respectivamente o número total de medições de temperatura de uma sequência obtida pelo sensor, e o tamanho dos intervalos, em minutos, em que as médias devem ser calculadas. As *N* linhas seguintes contêm um número inteiro cada, representando a sequência de medidas do sensor. O final da entrada é indicado quando *N = M =* 0.

**Saída de dados:**

Para cada conjunto de teste da entrada seu programa deve produzir três linhas. A primeira linha identifica o conjunto de teste, no formato “Teste *n*”, onde *n* é numerado a partir de 1. A segunda linha deve conter dois números inteiros *X* e *Y*, separados por ao menos um espaço em branco, representando respectivamente os valores da menor e da maior média de temperatura, conforme determinado pelo seu programa. O valor da média deve ser truncado, se a média não for um número inteiro (ou seja, deve ser impressa apenas a parte inteira). A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente. A saída deve ser gravada no arquivo “**lua.out**”

**Exemplo:**

|  |  |
| --- | --- |
| **Entrada (lua.in)** | **Saída (lua.out)** |
| 4 2  -5  -12  0  6  7 4  35  -35  5  100  100  50  50  0 0 | Teste 1  -8 3  Teste 2  26 75 |

**Restrições:**

0 ≤ *N* ≤ 10000 (*N* = 0 apenas para indicar o fim da entrada)

-200 ≤ *Temperatura* ≤ 200  
1≤*M*≤*N*

**Questão 3 (1 pt.):**

Quando limpavam o porão da casa recentemente herdada, os primos João e José descobriram um antigo mapa guardado no baú́ que havia sido de seu bisavô. O mapa parecia descrever uma ilha, era muito antigo, e em meio a indicações de caminhos pela ilha, continha apenas um nome: Huyn Chong Chong. Curiosos, João e José pesquisaram o nome na biblioteca do colégio e na Internet. Para sua surpresa e excitação, o nome era relacionado a uma antiga lenda de um tesouro escondido por piratas no século XVIII.

Encantados com a lenda, os primos acreditaram ter encontrado o mapa que os levaria ao tesouro, escondido na ilha de Huyn Chong Chong, próximo à Coreia do Sul. O tesouro, dizia a lenda, continha uma arca cheia de pedras preciosas muito raras e valiosas. Certos de que encontrariam o tesouro, os primos embarcaram rumo à ilha. Cada um dos primos se imaginava mais esperto do que o outro, e acreditava que encontraria o tesouro primeiro. Assim, eles combinaram que cada um ficaria com a parte do tesouro que encontrasse. Os primos então se separaram, e começaram a procurar o tesouro, especialmente a arca. Cada um dos primos tomou o caminho que imaginava que o levaria até a arca, e seguindo a indicação do mapa, ambos foram encontrando várias joias pelo caminho. Coincidentemente, os dois primos chegaram ao mesmo tempo no local onde a arca estava escondida. Como os dois encontraram a arca ao mesmo tempo, eles tinham agora que decidir como dividir o tesouro. Depois de analisar algumas alternativas, os primos concordaram em fazer a divisão da seguinte forma. Cada um ficaria com a parte do tesouro que encontrou antes de chegar à arca, e o conteúdo da arca seria dividido de forma que os dois ficassem com partes do tesouro total de mesmo valor. Para fazer a divisão desta forma, ao chegar de volta ao Brasil, os primos mandaram avaliar cada joia do tesouro. Contudo, eles estão agora em dúvida se é possível fazer a divisão conforme eles haviam combinado. Você̂, como amigo dos dois primos (agora milionários), e esperando receber alguma recompensa, dispôs-se a ajudá-los a descobrir se é possível fazer tal divisão.

**Tarefa:**

São dados:

* o valor dos objetos coletados por João e por José antes de encontrarem a arca;
* uma lista de valores, correspondentes aos objetos encontrados dentro da arca.

Como as joias são muito valiosas, estes valores são dados em unidades de R$ 1.000,00, ou seja, o valor 10 significa R$ 10.000,00. Você deve escrever um programa que determina se é possível dividir os objetos da arca de forma que, considerados também os valores dos objetos encontrados anteriormente (que ficarão com quem os encontrou), os primos recebam partes do tesouro com o mesmo valor.

**Entrada de dados:**

Seu programa deve ler vários conjuntos de testes, gravados no arquivo “**tesouro.in**”. A primeira linha de um conjunto de testes contém três números inteiros X, Y e N. Os valores X eY representam respectivamente a soma dos valores encontrados por João e por José antes de chegarem à arca. O valor N indica o número de objetos encontrados na arca. Seguem-se N linhas, cada uma contendo um número inteiro V, cor- respondendo ao valor de um dos objetos da arca. O final da entrada é indicado por X = Y = N = 0.

**Saída de dados:**

Para cada conjunto de teste da entrada seu programa deve produzir três linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato “Teste n”, onde n é numerado a partir de 1. A segunda linha deve conter o caractere ‘S’ caso seja possível dividir o tesouro como combinado pelos dois primos, ou o caractere ‘N’ caso contrário. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de saída, abaixo, deve ser seguida rigorosamente. A Saída deve ser gravada no arquivo “**tesouro.out**”.

**Exemplo:**

|  |  |
| --- | --- |
| **Entrada (tesouro.in)** | **Saída (tesouro.out)** |
| 10 20 4  3  8  7  2  1 1 6  2  7  7  12  5  3  0 0 0 | Teste 1  S  Teste 2  N |

**Restrições:**

0 ≤ X ≤ 50 (X = 0 apenas para indicar o final da entrada)

0 ≤ Y ≤ 50 (Y = 0 apenas para indicar o final da entrada)

0 ≤ N ≤ 100 (N = 0 apenas para indicar o final da entrada)

1 ≤ V ≤ 100

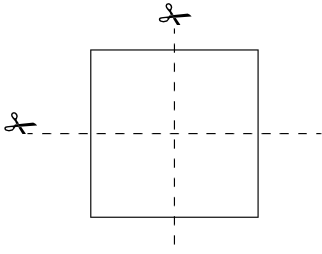
**Questão 4 (1 pt.):**

Zezinho tem aulas de Iniciação Artística em sua escola, e recentemente aprendeu a fazer dobraduras em papel. Ele ficou fascinado com as inúmeras possibilidades de se dobrar uma simples folha de papel. Como Zezinho gosta muito de matemática, resolveu inventar um quebra-cabeça envolvendo dobraduras. Zezinho definiu uma operação de dobradura D que consiste em dobrar duas vezes uma folha de papel quadrada de forma a conseguir um quadrado com 1/4 do tamanho original, conforme ilustrado na figura.

Forma, Retângulo

Descrição gerada automaticamente

Depois de repetir N vezes esta operação de dobradura D sobre o papel, Zezinho cortou o quadrado resultante com um corte vertical e um corte horizontal, conforme a figura abaixo.



Zezinho lançou então um desafio aos seus colegas: quem adivinha quantos pedaços de papel foram produzidos?

**Entrada:**

A entrada é composta de vários conjuntos de teste, gravados no arquivo “**dobra.in**”. Cada conjunto de teste é composto de uma única linha, contendo um número inteiro *N* que indica o número de vezes que a operação de dobradura D foi aplicada. O final da entrada é indicado por *N* = -1.

**Saída:**

Para cada conjunto de teste da entrada seu programa deve produzir três linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato “Teste *n*”, onde *n* é numerado a partir de 1. A segunda linha deve conter o número de pedaços de papel obtidos depois de cortar a dobradura, calculado pelo seu programa. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de saída, abaixo, deve ser seguida rigorosamente. A saída deve ser gravada no arquivo “**dobra.out**”.

**Exemplo:**

|  |  |
| --- | --- |
| **Entrada (dobra.in)** | **Saída (dobra.out)** |
| 1  0  -1 | Teste 1  9  Teste 2  4 |

**Restrições:**

-1 ≤ *N* ≤ 15 (*N* = -1 apenas para indicar o fim da entrada)

**Questão 5 (1 pt.):**

A crescente utilização do transporte aéreo preocupa os especialistas, que preveem que o congestionamento em aeroportos poderá́ se tornar um grande problema no futuro. Os números atuais já́ são alarmantes: relatórios oficiais demonstram que na Europa, em junho de 2001, houve uma média de 7.000 atrasos de voos por dia. Preocupada com a previsão dos seus especialistas em tráfego aéreo, a Associação de Transporte Aéreo Internacional (ATAI) está começando um estudo para descobrir quais são os aeroportos onde o tráfego aéreo pode vir a ser mais problemático no futuro.

**Tarefa:**

Como programador recém-contratado pela ATAI você foi encarregado de escrever um programa para determinar, a partir de uma listagem de aeroportos e voos, qual aeroporto possui maior probabilidade de congestionamento no futuro. Como medida da probabilidade de congestionamento será́ utilizado neste estudo o número total de voos que chegam ou que partem de cada aeroporto.

**Entrada:**

A entrada é composta de vários conjuntos de teste. A primeira linha de um conjunto de teste contém dois números inteiros *A* e *V*, que indicam respectivamente o número de aeroportos e o número de voos. Os aeroportos são identificados por inteiros de 1 a *A*. As *V* linhas seguintes contêm cada uma a informação de um voo, representada por um par de números inteiros positivos *X* e *Y*, indicando que há um voo do aeroporto *X* para o aeroporto *Y*. O final da entrada é indicado quando *A = V =* 0.

**Saída:**

Para cada conjunto de teste da entrada seu programa deve produzir três linhas, gravadas no arquivo “**aeroporto.in**”. A primeira linha identifica o conjunto de teste, no formato “Teste *n*”, onde *n* é numerado a partir de 1. A segunda linha deve conter o identificador do aeroporto que possui maior tráfego aéreo. Caso mais de um aeroporto possua este valor máximo, você deve listar todos estes aeroportos, em ordem crescente de identificação, e separados por pelo menos um espaço em branco. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de saída, abaixo, deve ser seguida rigorosamente.

**Exemplo:**

|  |  |
| --- | --- |
| **Entrada (aeroporto.in)** | **Saída (aeroporto.out)** |
| 5 7  1 3  2 1  3 2  3 4  4 5  3 5  2 5  3 5  1 3  1 2  3 2  1 2  2 1  0 0 | Teste 1  3  Teste 2  1 2 |

**Restrições:**

0 ≤ *A* ≤ 100 (*A* = 0 apenas para indicar o fim da entrada)

0 ≤ *V* ≤ 10000 (*V* = 0 apenas para indicar o fim da entrada)

1 ≤ *X* ≤ A

1 ≤ *Y* ≤ A

*X* ≠*Y*