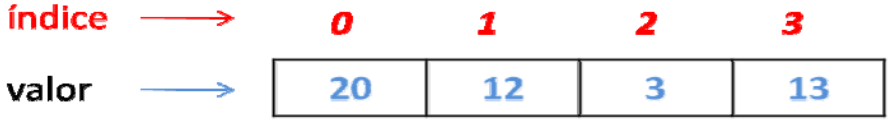
Roteiro 5 - Vetores

Até agora vimos variáveis capazes de armazenar **somente um valor** a cada momento, porém em diversas aplicações existe a necessidade de armazenamento de uma grande quantidade de valores ao mesmo tempo.

Vetores são **variáveis compostas homogêneas unidimensionais** capazes de armazenar vários valores de um mesmo tipo primitivo em um mesmo momento.

Os valores que compõem um vetor são identificados pelo mesmo nome, sendo diferenciado apenas por um índice. Além disso, os índices utilizados para identificar as posições de um vetor em JAVA **começam em 0 (zero) e vão até o tamanho o vetor menos uma unidade.**

Exemplo: Vetor com 4 posições. Inicia com índice 0 (zero) e termina com índice 3.



Portanto, para acessar os elementos do vetor é necessário utilizar o nome do vetor juntamente com o índice a ser referenciado. Suponha, por exemplo, que o identificador (nome) do vetor acima seja **peso**, e o programador quer fazer referência à segunda posição dele. O comando a ser realizado é **peso[1],** o qual retornará a o valor 12**.**

Declaração de um vetor

A declaração de um vetor é semelhante ao procedimento realizado para variáveis simples.

tipo [ ] nome = new tipo [tamanho_do_vetor];
ou
tipo nome [ ] = new tipo [tamanho_do_vetor];

tipo: indica o tipo de cada elemento do vetor
nome: é o identificador da variável
tamanho_do_vetor: indica o tamanho do vetor

Exemplos:

```
public class exemplo{
   public static void main( String [ ]args){
      int peso[ ] = new int[10];
      float [ ]nota = new float[40];
      String nome[ ] = new String [50];
   }
}
```


Elaboração Prof. Bruno Nonato Gomes

No exemplo acima foram criados 3 vetores: o primeiro com identificador **peso** de 10 posições que é capaz de armazenar valores inteiros; o segundo com nome **nota** de 40 posições que é capaz de armazenar valores reais; o terceiro com identificador **nome** de 50 posições que é capaz de armazenar Strings.

Atribuindo valores à um vetor

- O vetor, tal como variáveis simples, pode ser inicializado no momento da declaração.

Exemplo:

float nota[ ] = {4.5, 40, 90, 65.6}; // cria-se um vetor nota com 4 posições preenchidas na inicialização

- Um vetor também pode ser preenchido posição por posição utilizando o comando direto de atribuição.

Exemplo: preenchendo o vetor peso do exemplo acima.

```
peso[0] = 60;
peso[1] = 55;
.....
peso[9] = 78;
```

Porém, essa não é uma tarefa praticável. Imagine preencher um vetor de 1000 posições dessa forma?!

- Para preenchimento de um vetor de forma mais prática utiliza-se estrutura de repetição.

```
float nota[]=new float[40];
Scanner teclado = new Scanner(System.in);
for (int i =0; i < 40; i++){
   sout (" Digite o valor da posicao"+ i);
   nota[i] = teclado.nextFloat();
}
```

```
float nota[]=new float[40];
Scanner teclado = new Scanner(System.in);
for (int i =0; i < nota.length; i++){
   sout (" Digite o valor da posicao"+ i);
   nota[i] = teclado.nextFloat();
}
```

Estrutura de repetição for, pois podemos garantir que variável i assuma todos os possíveis valores para o índice do vetor. Em cada execução da estrutura de repetição, uma posição do vetor será preenchida. Vê-se que no segundo quadro, utilizamos **nota.length (**retorna o tamanho do vetor) como parada do for.

Imprimindo um vetor

De forma semelhante ao procedimento de atribuição, o vetor pode ser impresso posição por posição diretamente.

Exemplo: imprimindo vetor peso do exemplo acima.

```
cout<<peso[0];
cout<<peso[1];
.....
cout<<peso[9];
```

Porém, essa não é uma tarefa praticável. Imagine imprimir um vetor de 1000 posições dessa forma?!

Para impressão de um vetor de forma mais prática utiliza-se estrutura de repetição. Suponha o vetor nota com 4 posições, preenchido assim: float nota[ ] = {4.5, 40, 90, 65.6};

```
System.out.print("Vetor = [ ");
for (int i =0; i <4; i++) {
    System.out.print(nota[i] +" ");
}
System.out.println("] ");
```

Impressão:
Vetor = [4.5 40 90 65.6]

```
for (int i =0; i <4; i++) {
    System.out.print("Na posição "+ i +" temos o valor "
    + nota[i]);
}
```

Impressão:
Na posição 0 temos o valor 4.5
Na posição 1 temos o valor 40
Na posição 2 temos o valor 90
Na posição 3 temos o valor 65.6

Exemplos

Exemplo 1: Digite e compile o código abaixo. Nesse exemplo, faz-se o preenchimento e impressão de um vetor de notas com 5 posições.

```
1  package javaapplication1;
2  import java.util.Scanner;
3  public class JavaApplication1 {
4      public static void main(String[] args) {
5          float []notas = new float [5];
6          Scanner teclado = new Scanner(System.in);
7          for(int i=0; i<notas.length;i++){
8              System.out.println("Digite a nota da posição "+i+":");
9              notas[i] = teclado.nextFloat();
10         }
11         System.out.println("Impressão dos elementos do vetor.");
12         for(int i=0; i<notas.length;i++){
13             System.out.println("Nota["+i+"] = "+notas[i]);
14         }
15     }
16 }
```

Exemplo 2: Digite e compile o exemplo abaixo. Nesse exemplo, faz-se o preenchimento do vetor de notas e imprime a soma e média de todas as notas. Além disso, imprime as notas maiores ou iguais que a média geral.

```
1   package javaapplication1;
2   import java.util.Scanner;
3   public class JavaApplication1 {
4       public static void main(String[] args) {
5           float []notas = new float [5];
6           float soma = 0, media;
7           Scanner teclado = new Scanner(System.in);
8           for(int i=0; i<notas.length;i++){
9               System.out.println("Digite a nota da posição "+i+":");
10              notas[i] = teclado.nextFloat();
11              soma += notas[i];
12          }
13          media = soma/notas.length;
14          System.out.println("Soma das notas: "+soma);
15          System.out.println("Media das notas: "+media);
16          for(int i=0; i<notas.length;i++){
17              if(notas[i]>=media){
18                  System.out.println("Nota["+i+"] = "+notas[i]+" é maior ou igual à média");
19              }
20          }
21      }
22  }
```

O mesmo exemplo, porém utilizando o "for otimizado" (foreach).

```
1   package javaapplication1;
2   import java.util.Scanner;
3   public class JavaApplication1 {
4       public static void main(String[] args) {
5           float []notas = new float [5];
6           float soma = 0, media;
7           Scanner teclado = new Scanner(System.in);
8           for(int i=0; i<notas.length;i++){
9               System.out.println("Digite a nota da posição "+i+":");
10              notas[i] = teclado.nextFloat();
11              soma += notas[i];
12          }
13          media = soma/notas.length;
14          System.out.println("Soma das notas: "+soma);
15          System.out.println("Media das notas: "+media);
16          for(float valor:notas){
17              if(valor>=media){
18                  System.out.println("Nota = "+valor+" é maior ou igual à média");
19              }
20          }
21      }
22  }
```

Exercícios

1. Uma turma possui **40 alunos**. Faça um programa que leia o nome e a idade de todos os alunos e logo em seguida imprima:
   A. Total de alunos com idade menor ou igual a 16 anos
   B. Total de alunos com idade maior que 16 anos
   C. Média das idades
   D. Alunos com idade acima da média
   E. Nome e Idade do aluno mais novo
   F. Nome e Idade do aluno mais velho

2. Faça um programa que solicite o preenchimento de um vetor com 10 posições chamado **VetorOriginal**. Em seguida, o programa deve armazenar os números digitados de maneira invertida em vetor chamado **VetorInvertido**.

VetorOriginal

10	15	9	14	2	3	6	8	20	11

VetorInvertido

11	20	8	6	3	2	14	9	15	10

3. Faça um programa em JAVA que carregue em um vetor os N primeiros valores da serie de Fibonacci. Fibonacci = [1, 1, 2, 3, 5, 8, 13, 21, ...].

4. Faça um programa em JAVA que gere um vetor com 10 posições onde cada elemento coresponde ao quadrado de sua posição. Imprima o vetor resultante.

5. Em uma competição somente atletas que possuem altura maior ou igual a média geral das alturas dos inscritos podem participar. Sabe-se que a organização recebeu 20 inscrições. Faça um programa que identifique e mostre quais atletas podem participar da competição.

6. Leia um conjunto de N números inteiros (N<=100). Faça a divisão destes números em dois outros conjuntos seguindo a regra: Conjunto 1 – Apenas números positivos e pares e Conjunto 2 – Apenas números ímpares e/ou negativos.

7. Fazer um algoritmo que:
   a. Leia um conjunto de **valores inteiros** correspondentes a 80 notas dos alunos de uma turma, notas estas que variam de 0 a 10 (valide as entradas);
   b. Calcule a frequência absoluta e a frequência relativa de cada nota;
   c. Imprima uma tabela contendo os valores das notas (de 0 a 10) e suas respectivas frequências absoluta e relativa.

8. Construa um programa JAVA que preenche dois vetores reais de 10 posições, depois crie um terceiro vetor cujo conteúdo de cada posição é: 1, se o número armazenado em uma posição do vetor é o mesmo armazenado na posição respectiva do 2º, e 0, caso contrário.

9. Escreva um programa que leia um conjunto de 50 fichas correspondente à alunos e armazene-as em vetores, cada uma contendo, a altura e o código do sexo de uma pessoa (código = 1 se for masculino e 2 se for feminino), e calcule e imprima:
   - A maior, menor e a médias das alturas da turma;
   - As mulheres com altura acima da média da altura dos homens;
   - Os homens com altura abaixo da média da altura das mulheres;
   - As pessoas com altura abaixo da média da turma.

10. Faça um programa que preencha um vetor A de 10 posições e calcule o valor de S da seguinte forma:

$$S = (a[0] - a[9])^2 + (a[1] - a[8])^2 + (a[2] - a[7])^2 + (a[3] - a[6])^2 + (a[4] - a[5])^2$$

11. Faça um programa que carregue dois vetores de dez elementos e mostre um terceiro vetor resultante da intercalação desses dois vetores.

12. Faça um programa que preencha um vetor com 10 números digitados pelo usuário e em seguida crie e mostre um vetor ordenado de forma crescente e outro vetor ordenado de forma decrescente.

13. Intercalação é o processo utilizado para construir uma tabela ordenada, de tamanho n+m, a partir de duas tabelas já ordenadas de tamanhos n e m. Por exemplo, a partir das tabelas A = {1,3,6,7} e B = {2,4,5}, construímos a tabela C = {1,2,3,4,5,6,7}. Faça um algoritmo que:
   a. Leia NA, número de elementos do conjunto A (NA<=100);
   b. Leia, em seguida, os elementos do conjunto A;
   c. Leia, após o valor de NB, número de elementos do conjunto B(NB<=100);
   d. Leia, finalmente, os elementos do conjunto B;
   e. Crie e imprima um conjunto C, ordenado, de tamanho NA + NB, a partir dos conjuntos originais A e B.
   Obs: Considere os elementos de A e B como inteiros e já ordenados.

14. Faça um algoritmo em JAVA que preencha dois vetores de 10 posições e exiba um terceiro vetor resultante em que cada posição é a soma das posições dos outros vetores. Exemplo

```
vet1 = [10  27  3  -4   -6  -19  6   9  14   53 ]
vet2 = [  0  -7  6   4   -5   15  -6  19   1   47 ]
vet3 = [10  20  9   0  -11   -4   0  28  15  100 ]
```

15. Faça um programa que carregue um vetor de 5 posições, calcule e mostre o somatório e o produtório do mesmo.

16. Escreva um algoritmo que leia preencha e mostre um vetor de 10 posições, cada posição é composta pelo fatorial da mesma.

```
vet = [0  1  2  6   24  120  720   5040  40320   362880]
```

17. Desenvolva um programa que identifique, armazene em um vetor e mostre os 100 primeiros números primos a partir de 1. O algoritmo fica muito mais eficiente utilizando o crivo de Eratóstenes. Como

desafio, implemente o algoritmo utilizando a "forma tradicional" e depois utilizando o crivo de Eratóstenes, compute os tempos de ambas implementações e verifique a diferença entre os mesmos.

18. O desvio padrão de uma amostra de dados calcula o quanto de variação existe da amostra em relação à média. Valores baixos indicam que os dados tendem a estar próximos à média, por outro lado, valores altos indicam maior dispersão dos dados. O gerente de produção da Refrigerator Tabajará está avaliando o processo de fabricação de seus refrigerantes em lata, especificamente a etapa de preenchimento do conteúdo. As latas devem possuir 390 ml, com desvio padrão de + ou – 5 ml. Na análise foram coletadas 20 amostras aleatórias da linha de produção, medindo-se a quantidade de produto das mesmas. Você é o programador da empresa, e ficou incumbido de implementar um programa que auxilie seu gerente, mostrando ao mesmo a média e o desvio padrão entre os elementos das amostras. Além disso, exiba se o processo deve ou não ser revisado.

$$s = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n - 1}}$$

19. Implemente um algoritmo que preencha com valores inteiros um vetor de 10 posições. Em seguida solicite o usuário que digite um valor qualquer e pesquise se o valor informado pelo usuário pertence ou não ao vetor preenchido inicialmente. Caso não, mostre a mensagem "O valor X não está presente no vetor"; caso sim, mostre a mensagem "O valor X está na posição Y do vetor". Exemplo:

Vetor preenchido:	50	60	15	22	33	84	100

Valor digitado pelo usuário: 30 -> **Mensagem:** "Valor 30 não está presente no vetor"
Valor digitado pelo usuário: 84 -> **Mensagem:** "Valor 84 está na posição 5 no vetor"

20. Faça um programa que simule um controle bancário. O sistema bancário tem a capacidade de armazenar as informações somente de 50 usuários e contém as opções de cadastrar usuário; efetuar depósito; efetuar saque; consultar saldo em conta e finalizar o programa. O seguinte menu deverá aparecer o seguinte menu na tela para o usuário: (Desafio: fazer com interface gráfica)

Menu de Operações
1. Efetuar depósito
2. Efetuar saque
3. Consultar saldo em conta;
4. Finalizar o programa

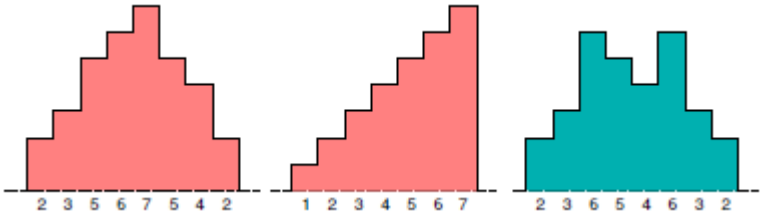
- Na opção de cadastrar usuário devem ser lidos os seguintes dados: nome do usuário (vetor de String com 50 posições); número da conta (vetor inteiro de 50 posições); saldo da conta (vetor float com 50 posições). O número da conta devem ser armazenados em um vetor de números inteiros (não pode haver mais de uma conta com o mesmo código) e os saldos devem ser armazenados em um vetor de números reais. O saldo deverá ser cadastrado na mesma posição do código. Por exemplo, se a conta 504 foi armazenada na quinta posição do vetor de número de contas, seu saldo deverá ficar na quinta posição do vetor de saldos.
- Para efetuar o depósito, deve-se solicitar o número da conta e o valor a ser depositado. Se a conta não estiver cadastrada, deverá aparecer a mensagem "Conta não encontrada!" e voltar ao menu inicial. Se a conta existir, atualizar e exibir o novo saldo;

- Para efetuar saque, deve-se solicitar o código da conta e o valor a ser sacado. Se a conta não estiver cadastrada, deverá aparecer a mensagem "Conta não encontrada!" e voltar ao menu. Se a conta existir, verificar se o seu saldo é suficiente para cobrir o saque. (Suponha que a conta não possa ficar com saldo negativo). Se o saldo for suficiente, realizar o saque, exibir o novo saldo e voltar ao menu. Caso contrário, mostrar a mensagem "Saldo insuficiente!" e voltar ao menu.
- Para consultar saldo em conta, deve-se solicitar o número da conta a ser pesquisada. Se a conta não estiver cadastrada, deverá aparecer a mensagem "Conta não encontrada!"; senão, mostrar a conta com seu respectivo saldo e voltar ao menu;

O programa termina quando for digitada a opção 4 – Finalizar o programa.

EXERCÍCIOS ESTILO OLIMPÍADAS

21. Um sistema de informações geográficas computadorizado está representando o perfil de uma montanha através de uma sequência de números inteiros, na qual não há dois números consecutivos iguais, como ilustrado na figura abaixo para três montanhas. Os números representam a altura da montanha ao longo de uma certa direção.



O gerente do sistema de informações geográficas pesquisou e encontrou uma maneira de identificar se uma sequência de números inteiros representa uma montanha com mais de um pico, ou com apenas um pico. Ele observou que, como não há números consecutivos iguais, se houver três números consecutivos na sequência, tal que o número do meio é menor do que os outros dois números, então a montanha tem mais de um pico. Caso contrário, a montanha tem apenas um pico. De forma mais rigorosa, se a sequência é A = [A1;A2;A3; ...; AN], ele quer saber se há uma posição i, para 2 <= i <= N - 1, tal que $A_{i-1} > A_i$ e $A_i < A_{i+1}$.
Para ajudar o gerente, seu programa deve determinar, dada a sequência de números inteiros representando a montanha, se ela tem mais de um pico, ou se tem um pico apenas.

Entrada

A primeira linha da entrada contém um inteiro N, representando o tamanho da sequência. A segunda linha contém N inteiros Ai, 1 <=1 i <= N, representando a sequência de alturas da montanha.

Saída

Seu programa deve imprimir uma linha contendo o caractere "S" se há mais de um pico, ou o caractere "N" se há apenas um pico.

Exemplos

Entrada	Saída
8 2 3 5 6 7 5 4 2	N

Entrada	Saída
8 2 3 6 5 4 6 3 2	S

22. Um experimento biológico utiliza uma fita de papel branco especial, na qual algumas gotas de um reagente são colocadas em posições específicas. Inicialmente a gota de reagente faz com que o papel se torne preto na posição em que foi colocada. A cada dia o reagente se propaga pelo papel, em todas as direções, com velocidade de 1 posição por dia, colorindo a região em que o reagente se propagou. A figura abaixo mostra um experimento com uma fita de 13 posições, com três gotas de reagente inicialmente, colocadas nas posições 2, 6 e 13 (a posição 1 é a primeira mais à esquerda da fita). Ao final do terceiro dia, a fita está completamente tomada pelo reagente.

Fita no estado inicial

Fita após um dia

Fita após dois dias

Fita após três dias

Você foi contratado para escrever um programa que, dados o comprimento da fita de papel e as posições das gotas de reagente no início do experimento, determine quantos dias serão necessários para a fita de papel ficar completamente tomada pelo reagente.

Entrada
A primeira linha contém dois inteiros F e R, indicando respectivamente o comprimento da fita de papel, em números de posições, e o número de gotas no início do experimento. A segunda linha contém R inteiros, indicando as posições das gotas de reagente, que são dadas em ordem crescente.

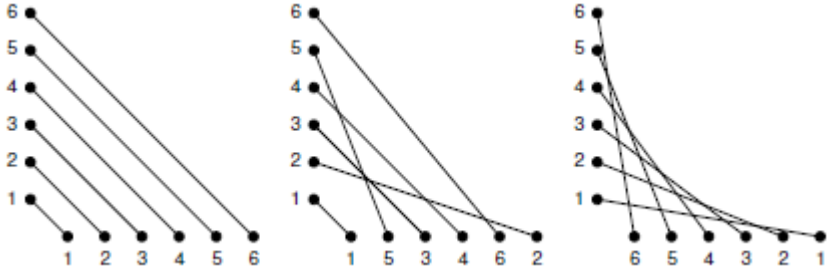
Saída
Seu programa deve produzir uma única linha, contendo um único inteiro, o número de dias necessários para que a fita de papel fique totalmente tomada pelo reagente.

Exemplos

Entrada	Saída
13 3 2 6 13	3

Entrada	Saída
10 2 9 10	8

23. Uma das atividades de recreação preferidas de Letícia é compor desenhos com linhas coloridas esticadas entre preguinhos numa base de madeira. Quanto mais cruzamentos entre pares de linhas, mais interessante fica a figura. Neste problema temos N pregos na vertical e N pregos na horizontal, como na figura abaixo. Os pregos na vertical possuem uma numeração fixa, de 1 a N, de baixo para cima. Os pregos na horizontal também são numerados de 1 a N, mas a ordem pode ser qualquer uma. Letícia vai sempre esticar uma linha entre cada par de pregos que tiverem o mesmo número. Dada a ordem dos pregos horizontais, seu programa deve computar o número total de cruzamentos entre pares de linhas no desenho de Letícia. Por exemplo, os três desenhos da figura possuem, respectivamente, 0, 6 e 15 cruzamentos.



Entrada
A primeira linha da entrada contém um número natural N. A segunda linha contém N números naturais distintos de 1 a N, representando a ordem dos pregos na horizontal.

Saída
Seu programa deve escrever uma linha na saída, contendo o número de cruzamentos entre pares de linhas, conforme a descrição anterior.

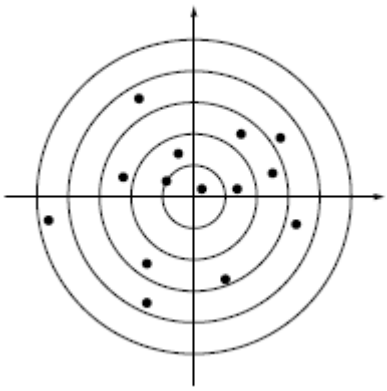
Exemplos

Entrada	Saída
6 1 5 3 4 6 2	6

Entrada	Saída
15 5 8 15 12 2 1 9 7 4 11 14 10 3 6 13	49

24. O comitê olímpico está testando uma nova forma de pontuar as competições de arco e flecha, baseada em penalidades. O atleta vai atirar N flechas no alvo, em sequência. A penalidade da K-ésima flecha atirada é computada imediatamente após ela atingir o alvo, antes do próximo lançamento, e é igual ao número de flechas que estão no alvo naquele momento cuja distância ao centro do alvo é menor ou igual à distância da K-ésima flecha ao centro, excluindo a própria K-ésima flecha. Quer dizer, a penalidade é o número das K - 1 flechas lançadas antes da K-ésima flecha que estão mais próximas ou à mesma distância do centro do alvo, comparadas com a K-ésima flecha. A penalidade total é a soma das penalidades das N flechas. Ganha o atleta que tiver a menor penalidade total ao final. Veja que a penalidade total pode ser zero, se o atleta for bom o bastante para acertar numa sequência estritamente decrescente de distâncias ao centro do alvo.

Neste problema, o centro do alvo está na origem (0; 0). Dada a sequência de coordenadas dos pontos em que as sucessivas flechas atingiram o alvo, seu programa deve computar a penalidade total final do atleta.



Entrada

A primeira linha da entrada contém um inteiro N, representando a quantidade de flechas lançadas. Cada uma das N linhas seguintes contém dois inteiros, X e Y , indicando as coordenadas do ponto em que cada flecha atingiu o alvo, definindo a sequência de lançamentos.

Saída

Imprima uma linha contendo um inteiro representando a penalidade total do atleta.

Exemplos

Entrada	Saída
2 1 3 5 4	1

Entrada	Saída
4 -100 85 -25 -60 18 33 0 0	0

Entrada	Saída
6 1 1 2 2 2 2 3 3 3 3 3 3	15