

Exercícios de Algoritmos

Prof. Eduardo Rosalém Marcelino

Última atualização: 05/2014

Livros que temos em nosso acervo com bons exercícios de programação, além de conteúdo:

- LIV 005.133 A839L ASCENCIO, Ana Fernandes Gomes. Lógica de programação com Pascal. São Paulo: Pearson Education do Brasil, 1999. 108 p. ISBN:8534610630.
- José augusto N. Z. manzano - Lógica de programação

Exercícios:

1. Faça um programa que leia as seguintes informações:

Nome de um aluno

Endereço

Telefone.

Depois de ler essas informações, exiba todas novamente em vídeo.

2. Delegacia

Solicite os seguintes dados: nome, sobrenome, rua, nº, bairro, cidade, estado, cep.

Exiba:

Sr: Nome + " " + Sobrenome + Residente à Rua, nº, bairro, cidade - estado, CEP. Favor comparar à delegacia.

3. Calcule a seguinte fórmula: (C# e fluxograma)

$X = b / m$.

Se $m = 0$, você não deve efetuar a divisão. E deve exibir um erro na tela.

Ao final, mostre o valor de x.

4. Programa faixa etária: (C# e fluxograma)

Solicite o nome e a idade de uma pessoa.

Então, exiba:

Olá _____ ! Sua idade é _____

Exiba também:

Se idade = 0 "bebê"

Se idade > 0 e idade <= 12 "Criança"

Se idade > 12 e idade <= 18 "adolescente"

Se idade > 18 e idade <= 60 "adulto"

Se idade > 60 e idade <= 120 "idoso"

Se idade > 120 "múmia"

5. Dia da Semana

Solicite um dia da semana (número inteiro) e seu programa deverá informar por extenso qual é o dia. EX: 1 = domingo.

6. Ano bissexto

Solicite ao usuário que digite um ano. Seu programa deverá informar se o ano digitado é bissexto ou não. Lembre-se que ano bissexto (de forma simplificada) é aquele que é divisível por 4 com resto igual a zero.

7. Saldo

Solicite:

nome

salário

despesas com saúde (ds)

despesas com lazer (dl)

despesas com alimentação (da)

saldo = salario - (ds+dl+da)

se saldo <=0 "coitado!"

se saldo entre 0 e 3000 "guarde para o mês que vem"

se saldo entre 3001 e 5000 "muito bom! está divorciado?"

se saldo > 5000 "parabéns corintiano"

8. Solicite 3 notas de um aluno.

Calcule a média:

$$m = (n1 + n2 + n3) / 3$$

Depois exibir a média e exibir a situação de acordo com as regras a seguir:

se $m < 4$ -> reprovado direto

se $m \geq 7$ -> aprovado

caso contrário, está de recuperação. Neste caso, solicite a nota de recuperação.

Se esta nota for < 5 -> reprovado

Se for ≥ 5 -> aprovado

9. MRU - movimento retilíneo uniforme. Calcula o tempo e o local de encontro de 2 móveis (C# e fluxograma)

Formula:

$$t = d_{ab} / (v_a - v_b)$$

Onde:

t = tempo

dab = distância entre o móvel a e o móvel b

va = velocidade do móvel a

v_b = velocidade do móvel b

$local = v_a * t$

Onde:

local = local (KM) do encontro

Exibir ao final:

Se os móveis se encontrarem ou não, e em caso de se encontrarem, em que tempo (t) e em que local

obs: se $v_a \leq v_b$ os móveis nunca se encontrarão!

10. Renda

Solicite: nome, salário, sexo(M/F) e exiba ao final:

Olá sr(a) nome, seu salário é ____

Seu sexo é:

(se for 'M' ou 'm' exibir "masculino", se for 'F' ou 'f' exibir "feminino")

(Se salario = 0 exibir "falido"

salario entre 1 e 15000 "marajá"

salario > 15000 "Corintiano"

11. Faça um programa que solicite seu nome completo, o CPF , a idade e o salário.

Após ler todos os valores, exiba:

A idade.

O primeiro nome do nome digitado.

As últimas 3 letras do nome digitado.

Calcule e exiba o desconto de INSS (11%)

Exiba o salário final já descontado o INSS

Exibir o CPF sem os pontos e sem traço.

12. Solicite o nome e o sexo de uma pessoa.

Exibir:

Se o nome da pessoa começar com uma vogal, exibir:

Olá _____ seu nome inicia com uma vogal.

Caso contrário, exiba:

Olá _____ seu nome inicia com uma consoante.

Se o sexo for 'M' ou 'm', exiba:

Sexo: Masculino

Se o sexo for 'F' ou 'f', exiba:

Sexo: Feminino

Se o sexo não for nenhuma das opções anteriores, exibir:

Sexo: Indefinido

13. Hora válida

Solicite uma hora no formato HH:MM (formato 24 horas) e informe se a hora é válida ou não.

14. Tabuada.

Solicite um número qualquer e calcule a tabuada deste número de forma que o resultado fique assim:

ex: para o nº 7

$$7 \times 1 = 7$$

$$7 \times 2 = 14$$

$$7 \times 3 = 21$$

$$7 \times 4 = 28$$

...

$$7 \times 10 = 70$$

15. Faça um programa para somar todos os números de 1 a 100 e exibir o resultado em vídeo.

16. Faça um programa que exiba os nºs pares e os ímpares de 1 a 1000.

Dica : número par é todo aquele cujo resto da divisão inteira por 2 dá ZERO.

% = operador de resto da divisão

Ex:

$$4\%2 == 0 \text{ resto ZERO (PAR)}$$

$$5\%2 == 1 \text{ resto 1 (IMPAR)}$$

17. Sabendo-se que o comando Writeline pula de linha após exibir um texto, faça um programa que imprima na tela um texto informado pelo usuário, letra por letra, cada uma em uma linha. Ex: texto = "vertical"

Ex:

v
e
r
t
i
c
a
l

18. FATORIAL

Faça um programa que solicite um número e exiba o fatorial deste número.

ex.

$$5! = 5 * 4 * 3 * 2 * 1 = 120$$

19. Conta Letras

Faça um programa que solicite uma frase e uma letra. Então, seu programa deverá exibir a quantidade de vezes que a letra aparece na frase.

Ex:

Digite uma frase: "Olá mundo cruel"

Digite uma letra: "l"

A letra [l] aparece 2 vezes na frase.

20. ContaLetraSemEspacos

O comando Length das variáveis do tipo string conta todas as letras de uma String, inclusive os espaços.

Faça um programa que conte as letras de uma string, ignorando os espaços. Exiba no vídeo o total de letras. Não use o replace!

21. Faça um programa que solicite um texto e então exiba-o invertido:

ex:

BOZO

R: OZOB

22. Faça o programa da tabuada ao contrário:

Ex:

$$7 * 10 = 70$$

$$7 * 9 = 63$$

...

$$7 * 0 = 0$$

23. Faça um programa para preencher a sua tela com 1 caractere escolhido pelo usuário.

24. Vogais

Faça um programa que solicite um texto qualquer, e então seu programa deverá exibir em vídeo apenas as vogais do texto.

25. O programa abaixo está correto e não necessita de ajustes. Faça a indentação do código no bloco de notas ou no seu caderno. Descubra qual o valor que será exibido caso texto seja "Sardinha"

```
static void Main(string[] args)
{
    string texto = Console.ReadLine();
    int soma = 0;
    char x;
    for (int n = texto.Length - 1; n >= 0; n--)
    {
        x = texto.ToUpper()[n];
        if (x == 'A' || x == 'E')
            soma += 10;
        soma++;
        if (x == 'I' || x == 'O')
        {
            soma = soma + 20;
            if (n < texto.Length - 2)
                soma = soma + (soma % 2);
        }
        if (x == 'U')
            soma--;
    }
    Console.WriteLine(soma);
    Console.ReadLine();
}
```

26. O programa abaixo está correto e não necessita de ajustes. Faça a indentação do código no bloco de notas ou no seu caderno. Caso a matrícula digitada seja 5698741, qual será o valor exibido ao final do programa?

```
static void Main(string[] args)
{
    Console.WriteLine("Digite o número da matrícula composto " +
        "por 7 caracteres numéricos");
    string matricula = Console.ReadLine();
    if (matricula.Length == 7)
    {
        int soma = 0;
        for (int n = 0; n < matricula.Length; n++)
        {
            int valor = Convert.ToInt32(matricula.Substring(n, 1));
            soma = soma + valor;
        }
    }
}
```

```

}
int digito = soma % matricula.Length;
Console.WriteLine("Matricula com dígito: {0}",
matricula + "-" + digito.ToString());
}
else
{
Console.WriteLine("O texto digitado não possui 7 dígitos numéricos");
}
Console.ReadLine();
}

```

27. Refaça os programas Tabuada até Fatorial usando o do-while.

28. Refaça os programas Conta Letras até o programa Vogais usando o while.

29. Faça um programa de cadastro. Solicite os seguintes dados:

Cadastro de Detento

codigo : integer (maior que zero)
nome : string (Obrigatorio pelo menos 1 letra)
condenacao : string (obrigatorio ao menos 5 letras)
pena : int (entre 1 e 500 anos)

Ao final do cadastro, exiba os dados:

Codigo: _____
Nome: _____
Condenacao: _____
Pena: _____

Pergunte se o usuário deseja cadastrar outro detento ou sair do programa.

30. Faça um programa para uma loja de informática.

Solicite:

codigo : int <entre 0 e 500>
descricao : string - pelo menos 3 letras
tipo : char - H ou S (hardware ou software)
valor : double >= 0

Ao final exibir :

Codigo - descrição - valor - tipo (exibir Software quando S e Hardware quando H);
Tudo em uma única linha. Perguntar se o usuário deseja sair ou continuar.

31. PROGRAMA SOBRENOME;

Faça um programa que solicite um nome e após validar se o nome foi digitado completo (com nome e sobrenome), ele deverá exibir apenas o sobrenome. Caso não seja informado um sobrenome, exibir um erro. Não utilize o comando substring!

Ex:

Informe um nome: ana lucia da silva

Seu sobrenome é: lucia da silva

32. UltimaOcorrencia

O método indexof da classe string retorna a posição da primeira ocorrência de uma letra em um texto. Faça um programa que solicite um texto e uma letra (char) . Exiba no vídeo a última ocorrência desta letra no texto, ou -1.

33. Soma e Média

Faça um program que leia N valores e os acumule em uma variável. A cada valor lido, pergunte ao usuário se ele deseja parar ou continuar. Quando ele solicitar parar, exiba um menu com as seguintes opções:

- 1 - Exibir a soma dos valores lidos
- 2 - Exibir a média dos valores lidos
- 3 - Sair

34. Conta Bancária

Solicite os seguintes dados de uma conta bancária:

Saldo do cliente : Double

Limite de cheque especial: Double.

Então, seu programa deverá entrar em um loop, solicitando as seguintes opção ao cliente:

“D” depósito

“S” saque

“V” visualizar o saldo

“X” sair do programa

Caso o usuário tenha escolhido “D”epósito, de vê-se acrescentar o valor ao saldo do cliente.

Caso o usuário tenha escolhido “S”aque, de vê-se subtrair o valor do saldo do cliente. O saldo pode ficar negativo até o limite de cheque especial.

Caso o usuário tenha solicitado “V”isualizar o saldo, exiba o saldo no vídeo.

Caso o usuário tenha solicitado para sair do programa, saia do loop e termine a aplicação.

35. Solicite os seguintes dados de um jogo:

- String Nome
- Char Categoria (“A” para ação /aventura, “C” corrida , “L” Luta , “R” Rpg e “E”esportes
- String Data da compra
- Double Valor de Locação

Salve os dados ao final em um arquivo texto. Ao entrar no programa, caso o arquivo exista, leia seus valores e os exiba em vídeo.

36. Faça um programa que leia uma string e vá acumulando as strings digitadas até que o usuário digite “sair”. O programa deve persistir em disco todas as strings digitadas e fechar o programa assim que for digitado “sair”. Ao abrir o programa novamente, caso o arquivo com os dados exista, exiba todas as palavras digitadas na tela, uma por linha.

37. Notas musicais podem ser tocadas pelo C# utilizando-se o método Console.Beep. Sua sintaxe é: Console.Beep(nota, duração); sendo que nota é um valor inteiro que representa uma nota musical, e duração é um valor inteiro que representa a quantidade de milissegundos que a nota deverá ser tocada.

EX: Console.Beep(“261”, 1000) ; irá tocar a nota “Dó” por 1 segundo, ou seja 1000 milissegundos.

Com base nisso, faça um programa que execute as notas musicais definidas em uma string, como a do exemplo abaixo. Os valores inteiros representam as notas. Cada nota vem seguida de pontos que indicam o tempo que a nota será tocada. Cada “.” representa 100 milissegundos. Quando for encontrado o comando “|” deve ser tocado: `Console.Beep(37, 100) ;`

```
string notas =  
    "261....293....329...349...|349..|349...|" +  
    "261..293...261..293...|293..|293..|" +  
    "261....392....349....329...|329...|329...|" +  
    "261....293....329...349...|349..|349....|";
```

VETORES

38. Program CincoNomes;

Faça um programa que solicite 5 nomes e 5 salários e então exiba os dados no seguinte formato:

Nome - Salário

Ex:

José da Silva 1580,00

Maria Xavier 1600,00

...

39. Solicite 5 nomes e exiba-os de forma invertida.

40. Solicite 5 números e armazene-os em um vetor. Depois, exiba em vídeo qual foi o maior número lido.

41. MaiorMenor

Faça um programa que solicite 5 números. O programa deverá ao final exibir TODOS os números cadastrados no seguinte formato (utilize estrutura de Repetição):

n1 - n2 - n3 - n4 - n5

E também deverá informar qual foi o maior e o menor número cadastrado.

42. Solicite até 30 nomes. Após ler os 30 nomes, exiba um menu com as seguintes opções:

- 1-> exibir todos os nomes
- 2-> exibir o maior nome (que tem mais letras). Se, por exemplo, os 2 maiores nomes tiverem o mesmo número de letras, exibir os 2.
- 3-> exibir apenas o primeiro nome de todos os nomes informados
- 4-> Exibir apenas os nomes que inicial com vogal.
- 5-> Sair

43. VetNomSal

Faça um programa que permita cadastrar até 50 nomes e 50 salários.

Os nomes devem ser preenchidos e os salários devem estar entre R\$ 0 e 15000.

O usuário poderá parar o cadastro se quiser.

Após o cadastro, exiba um menu com as seguintes opções:

- 1-) Listar todos os cadastrados no formato: nome - salário.
- 2-) Exibir todas as ocorrências de quem tem o menor salário.
- 3-) Exibir só o sobrenome de todos os cadastrados. Se não houver sobrenome, exiba o primeiro nome.
- 4-) Exibir a média salarial
- 6-) Sair

44. FindNum

Faça um programa que solicite 10 números inteiros.

Depois o programa deverá entrar em uma repetição, onde ele solicitará um número qualquer e dirá se o número existe ou não no vetor previamente digitado.

Isso se repetirá até que o usuário queira sair.

45. NumIguar;

Faça um programa que solicite 10 números. O programa não deve permitir números repetidos!

46. Loteria

Crie um programa para gerar os 6 números da loteria (1 a 60). Cuidado para não gerar números repetidos!

Ao final, exiba os números gerados.

47. Multiplica Matriz

Faça um programa que multiplique 2 matrizes e exiba ao final a matriz de resultado.

EX:

```
| 2 3 |   | 3 1 |  
| 1 0 | x | 2 4 |  
| 4 5 |
```

Resposta: | 12 14 |
 | 03 01 |
 | 22 24 |

Para quem não se lembra de como multiplicar matrizes:

<http://www.infoescola.com/matematica/operacoes-com-matrizes-multiplicacao/>

<http://www.brasilecola.com/matematica/multiplicacao-matrizes.htm>

Para testar na web: <http://www.bluebit.gr/matrix-calculator/multiply.aspx>

Dicas na internet: <http://msdn.microsoft.com/en-us/library/aa288453%28v=vs.71%29.aspx>

CONTROLE DE EXCEÇÃO

48. Solicite o código, o salário e a data de nascimento de um aluno. Caso ele digite valores incorretos, solicite-os novamente. Utilize o controle de exceção. Ao final, mostre os valores.

ESTRUTURAS HETEROGÊNEAS

(Utilize controle de exceção nas soluções quando necessário)

49. Faça um programa para gravar os seguintes dados de um detento:

nome : string; Não pode estar vazio

pena : int ; Entre 1 e 500 anos

condenação : string ; Obrigatória pelo menos 5 letras

Ao final exiba os dados no formato nome - pena - condenação

50. Sistema Acadêmico

Faça um programa que solicite os seguintes dados de um aluno: nome, telefone, idade, valor da mensalidade.

Ao final do programa, exiba os dados cadastrados.

Validações:

Nome deve ser preenchido, telefone deve ter exatamente 9 dígitos (contando o "-")

Dígitos não deve ser aceito celular (números que iniciam com 7,8 ou 9)

Idade deve estar entre 0 e 150 e a mensalidade deve ser ≥ 0 .

51. Funcionários

Faça um programa que armazene até 30 funcionarios, guardando o nome e o salário.

Ao final, exiba: O maior salário, o menor salário, a soma dos salários e a média salarial.

52. Caixa

Faça um programa para armazenar o caixa de uma empresa.

O caixa é registrado diariamente informando se:

* DATA (DD/MM/AA) (para simplificar a validação: DD entre 1 e 31, MM entre 1 e 12 e AA > 00) com 8 posições.

* Total de Entradas (≥ 0)

* Total de Sidas (≥ 0)

Permita a digitação de um caixa de até 30 dias.

O usuário poderá parar a qualquer momento.

Ao termino da digitação, exiba um menu com as seguintes opções:

1-) Exibir o caixa no formato : DATA - Entradas - Saídas

2-) Exibir o Total geral de entradas

3-) Exibir o Total geral de saídas

4-) Exibir o saldo : Total de entradas - Total de Saídas

5-) Exibir o caixa por mês. Solicite o mês e exiba somente (MM) os dados pertinentes no mesmo formato da opção 1.

7-) Sair.

53. Caixa 24 horas.

Um caixa 24h hipotético possui em seu interior notas e moedas dos seguintes valores:

R\$ 0,05

R\$ 0,10

R\$ 0,25

R\$ 0,50

R\$ 1,00

R\$ 2,00

R\$ 5,00

R\$ 10,00

R\$ 20,00

R\$ 50,00

R\$ 100,00

Vamos supor também que a quantidade de cada um desses valores seja infinito, ou seja, o caixa tem muito dinheiro...

Faça um programa que o usuário digite o valor que ele deseja sacar no caixa, e o caixa deverá exibir em vídeo as notas/moedas que foram utilizadas para compor o valor solicitado, sempre priorizando as notas com valores mais altos primeiro.

Ex: Se o cliente pedir 125,30, deverá ser exibido em vídeo:

1 de 100
1 de 20
1 de 5
1 de 0,25
1 de 0,05

54. Automóvel

Crie uma estrutura (MODELO) para guardar o modelo de um automóvel:

string marca (obrigatório)
string tipo (obrigatório)

Crie outra estrutura (CARRO) para guardar os dados de um determinado automóvel:

string placa (obrigatório 3 letras, um hífen, 4 números: EX: XXX-9999)
int ano de fabricação >= 2000
MODELO modelo

Crie um vetor para cadastrar até 10 carros. Pergunte se o usuário deseja parar o cadastro após cadastrar um carro. Ao final, exiba os dados em vídeo.

MÉTODOS

(function = metodo que devolve
procedure = metodo void)

55. Primeiro Nome

Faça um programa que tenha um método que receba um nome completo ou não. Retorne então apenas o primeiro nome. No programa principal vc deverá solicitar o nome de 2 pessoas e executar este método.

```
static string RetornaPrimeiroNome(string Nomecompleto)
{
    return ???
}
```

56. Par Impar

Faça um programa que tenha uma método que receba um número e devolva um String dizendo se o número é Par ou Impar.

57. Numero Primo

Faça um programa que tenha uma método que receba um número e devolva um String dizendo se o número é PRIMO.

58. Vogais

Faça um programa que tenha uma método void que recebe um nome e o objetivo do método é escrever no vídeo apenas as vogais deste nome.

59. Nome Invertido

Faça um programa que solicite um texto qualquer e então um método chamado irá devolver o texto recebido como parâmetro invertido. ex: nariz r= ziran

60. Três Métodos

Faça um programa que tenha 3 métodos :

UltimaLetra -> recebe um texto e exibe a última letra dele.

TrocaCaracter -> recebe um texto, o caractere original e o novo caractere.

Ele substituirá todo os caracteres originais pelo novo caractere e deverá devolver o novo texto.

Iniciais -> recebe um texto e retorna suas iniciais.

Faça o programa principal de forma que se possa testar todas estes métodos

61. Verifica Espaço

Faça um método cujo objetivo é verificar se existe espaço (' ') em um texto informado via parâmetro. Ela deverá retornar True se houver espaço e False se não houver. Faça o programa principal para testar esta function.

62. Valida E-mail

Faça um método cujo objetivo é validar um e-mail informado via parâmetro. Ela deverá retornar TRUE se o e-mail for válido ou FALSE caso contrário. Um e-mail válido é aquele que possui um @ (arroba) e um . (ponto) em seu conteúdo, mas não inicia ou termina com eles.

63. Tabuada

Faça um método void que imprima a tabuada de um número. Faça o programa principal para testá-la.

64. Menor valor

Faça um método que receba um vetor de inteiros e ele deverá devolver qual o menor valor no vetor.

65. Soma

Faça um método que receba um vetor de inteiros e ele deverá devolver a soma dos elementos do vetor.

66. Media

Faça um método que receba um vetor de inteiros e ele deverá devolver a média dos elementos do vetor.

67. Faça um método que receba um vetor de inteiros e ele deverá devolver:

- a média dos elementos do vetor.
- a soma
- o maior valor
- o menor valor

68. Acho que tinha terminado?

Altere o programa 50. Crie um método para validar o telefone. Ela deverá receber como parâmetro o número do telefone e retornar True se o telefone estiver correto ou False caso contrário.

69. Crie um método que devolva 3 informações sobre um nome:

- A quantidade de vogais do nome
- O primeiro nome do nome informado
- Uma versão do nome com apenas a primeira letra de cada palavra em maiúsculo

DESAFIOS:

1. Validação de CPF

No Brasil existe o CPF (Cadastro de Pessoas Físicas) que serve para identificar cada indivíduo no país. O número do CPF é composto de 11 dígitos, sendo os dois últimos os dígitos de verificação. A fórmula para verificar a validade do número do CPF é simples e é explicada abaixo:

Vamos tomar como exemplo o número de CPF **123.456.789-09**

• Calculando o 1º Dígito Verificador

Primeiro calculamos a soma da multiplicação dos 9 primeiros dígitos por **10, 9, 8, ... , 3, 2**, respectivamente. Ou seja:

$$\text{Soma} = (1 \cdot 10) + (2 \cdot 9) + \dots + (8 \cdot 3) + (9 \cdot 2)$$

Depois calculamos o resto da divisão de soma por 11:

Resultado = Soma **módulo** 11.

Agora analisamos **Resultado**:

- Se **Resultado** for igual a **0** ou **1**, então o 1º dígito verificador deve ser **0**;
- Caso contrário, o 1º dígito verificador é calculado com a fórmula: **11 - Resultado**.

• Calculando o 2º Dígito Verificador

Primeiro calculamos a soma da multiplicação dos 9 primeiros dígitos por **11, 10, 9, ... , 4, 3**, respectivamente e em seguida somamos com **(Dígito1*2)**, sendo que **Dígito1** é o valor encontrado para o 1º dígito verificador. Ou seja:

$$\text{Soma} = (1 \cdot 11) + (2 \cdot 10) + \dots + (8 \cdot 4) + (9 \cdot 3) + (\text{Dígito1} \cdot 2)$$

Depois calculamos o resto da divisão de soma por 11:

Resultado = Soma **módulo** 11.

Agora analisamos **Resultado**:

- Se **Resultado** for igual a **0** ou **1**, então o 2º dígito verificador deve ser **0**;
- Caso contrário, o 2º dígito verificador é calculado com a fórmula: **11 - Resultado**.

No nosso exemplo (**123.456.789-09**) o número é válido.

Desafio:

Faça um programa que solicite um CPF e então seu programa deverá informar se o CPF é válido ou não, além de exibir os dígitos calculados.

Lembrando que o operador de módulo (resto de divisão inteira) em C# é o `%`.

2. Sequência

Dado um nº inicial, calcule N sequências, onde o usuário irá escolher o número e o valor de N.

EX:

1

11

21

1211

111221

312211

No exemplo acima há 6 sequências (linhas).

A lógica para criar uma nova sequência é a seguinte: com base na última, contam-se os caracteres iguais que aparecem em sequência. Então, coloca-se a quantidade de caracteres seguidos do caractere que estava sendo analisado. Sendo assim, no exemplo acima o valor da próxima linha, tomando como base a última linha gerada até então (312211) será:

(1 caractere 3, 1 caractere 1, 2 caracteres 2, 2 caracteres 1): 13112221.

Vale ressaltar que o primeiro elemento da sequência sempre será composto por apenas um caractere.

3. Aposta Milionária

O senhor Eduardo está muito a fim de ficar milionário ainda este ano. Para melhorar a estratégia de suas apostas, ele resolveu analisar os resultados de apostas realizadas anteriormente por diversos apostadores, isto é, seus “concorrentes”. As únicas informações que ele possui sobre essas apostas são a quantidade de dinheiro ganha ou perdida e a sequência em que elas ocorreram. Por exemplo, certo dia, um determinado apostador teve os seguintes resultados.

1	2	3	4	5	6	7	8	9	10	11
---	---	---	---	---	---	---	---	---	----	----

5	3	-10	7	8	-5	-4	9	0	1	-15
---	---	-----	---	---	----	----	---	---	---	-----

Isto é, na primeira aposta, o apostador ganhou 5. Na segunda, ganhou 3. Na terceira perdeu 10, e assim por diante. O Eduardo pretende analisar esses resultados e gostaria de identificar padrões de vitórias consecutivas, e, portanto descobrir uma subsequência da sequência de apostas na qual o valor ganho seja máximo. No exemplo acima, esta situação ocorre na subsequência que se inicia na aposta 4 e termina na aposta 10, e o ganho é $7 + 8 - 5 - 4 + 9 + 0 + 1 = 16$. Note, portanto, que a melhor subsequência pode incluir valores negativos, desde que a quantidade perdida seja compensada por apostas vencedoras.

Escreva um programa em C# que leia uma sequência de apostas e identifique uma subsequência de ganho máximo.

ENTRADA

A entrada será fornecida em um arquivo texto (entrada.txt) da seguinte forma: o primeiro valor lido corresponde a um inteiro positivo N que indica a quantidade de apostas na sequência; em seguida, devem ser lidos X números inteiros correspondentes aos valores das apostas. Considere que N é sempre maior que 0 e menor ou igual a 1000. Considere também que o valor de uma aposta (Y) está sempre entre -2147483648 e 2147483647. O término de leitura dos testes será dado quando N for igual a zero.

SAÍDA

A saída do programa deve ser composta de uma linha contendo três valores separados por espaços em branco: o valor do ganho máximo identificado, o número da aposta que inicia a sequência de ganho máximo e o número de aposta que finaliza a sequência. Se não houver uma subsequência com possibilidade de ganho de dinheiro, seu programa deve imprimir todos esses valores como zero. Possivelmente, existirá mais de uma subsequência na qual o ganho é máximo. Neste caso, a saída do seu programa deve imprimir aquela que aparece primeiro (menores índices de início e fim).

Exemplo de ENTRADA

```
11
5 3 -10 7 8 -5 -4 9 0 1 -15
8
-1 -3 0 -12 -8 0 0 -3
0
```

Exemplo de SAÍDA

```
16 4 10
0 0 0
```

Outro exemplo de entrada:

```
11
5 3 -10 7 8 -5 -4 9 0 1 -15
8
-1 -3 0 -12 -8 0 0 -3
9
1 2 3 5 7 9 -1 -20 7
12
-5 1 6 3 -2 10 -5 -8 6 -1 -7 10
8
1 2 3 -5 -1 1 2 3
6
-5 -6 20 -1 -10 -9
6
70 1929 2049 -5 -5000 10000
7
70 1929 2049 2000 -5 -5000 10000
10
70 1929 2049 2000 -5 -5000 10000 -30000 50000 50000
7
-10 -20 -30 -50 -1000 -5000 -3000
19
20 10 30 60 -200 60 50 -1000 6 7 8 9 -50 2500 -1 2760 60 90 -70
10
200 7 200 -50 -70 200 7 200 -30 -20
10
200 7 200 -50 -7000 200 7 200 -30 -20
0
```

E para essas entradas, a saída seria:

```
16 4 10
0 0 0
27 1 6
18 2 6
6 1 3
20 3 3
10000 6 6
11043 1 7
100000 9 10
0 0 0
5409 14 18
694 1 8
407 1 3
```

RECUPERAÇÃO

1) Ano bissexto

Solicite ao usuário que digite um ano. Seu programa deverá informar se o ano digitado é bissexto ou não. Lembre-se que ano bissexto (de forma simplificada) é aquele que é divisível por 4 com resto igual a zero.

2) Data completa

Solicite uma data completa no formato d/m/aaaa . O seu programa deverá dizer se a data informada é válida ou não. Considere ano bissexto, meses que tem 28,29 , 30 ou 31 dias e meses que vão de 1 a 12.

3) Dia da Semana

Solicite um dia da semana (número inteiro) e seu programa deverá informar por extenso qual é o dia. EX: 1 = domingo.

4) Hora válida

Solicite uma hora no formato HH:MM (formato 24 horas) e informe se a hora é válida ou não.

5) Número inteiro correto

Solicite um número. O seu programa deverá dizer se o número é válido ou não. Número inteiro válido é aquele que possui apenas números (sem espaços ou outros caracteres). O número pode ser negativo, o que significa que você deverá aceitar 1 sinal de menos '-' no início do número.

6) Primeira vogal

Faça um programa que solicite um texto e então você deverá exibir a primeira vogal deste texto.

7) Última vogal

Faça um programa que solicite um texto e então você deverá exibir a última vogal deste texto.

8) Manipulação de Matriz

Faça um programa que preencha uma matriz 5x5 com inteiros e exiba em vídeo:

- a-) a soma dos números ímpares
- b-) a soma de cada uma das colunas
- c-) a soma de cada uma das linhas

9) Soma

Faça um programa que solicite ao usuário uma string onde ele deverá digitar apenas números. Seu objetivo então é somar todos os números dessa string e exibir a soma.

EX: texto = "1235671". A soma será 25.