****

**Introdução à Ciência da Computação - 113913**

**Lista de Exercícios 4**

**Iteração, Estruturas de Repetição**

**Observações:**

* As listas de exercícios serão corrigidas por um **corretor automático,** portanto é necessário que as entradas e saídas do seu programa estejam conforme o padrão especificado em cada questão (exemplo de entrada e saída). Por exemplo, a não ser que seja pedido na questão, não use mensagens escritas durante o desenvolvimento do seu código como “Informe a primeira entrada”. Estas mensagens não são tratadas pelo corretor, portanto a correção irá resultar em resposta errada, mesmo que seu código esteja correto.
* As questões estão em **ordem de dificuldade**. Cada lista possui 7 exercícios, sendo 1 questão fácil, 3 ou 4 médias e 2 ou 3 difíceis.
* Leia com atenção e faça **exatamente** o que está sendo pedido.

1. Um pequeno jogo de adivinhação funciona da seguinte forma: você define um número ***n*** e chama um amigo, que deverá adivinhar o número escolhido. Faça um programa que peça um inteiro e então fique pedindo que um usuário tente adivinhá-lo até que acerte. Em cada tentativa o programa deve dizer se o chute foi maior ou menor que o número certo.

**Entrada**

A primeira linha de entrada o inteiro ***n,*** que deverá ser adivinhado. As próximas linhas serão os números chutados pelo jogador, que continuará chutando números até que adivinhe o número correto.

**Saída**

Se o número digitado for menor que ***n*** apresente a mensagem: “O número correto é maior.”. Se o número digitado for maior que ***n*** apresente a mensagem: “O número correto é menor.”. Quando o usuário acertar o número imprima: “Parabéns! Você acertou.”.

|  |  |
| --- | --- |
| **Exemplo de Entrada** | **Exemplo de Saída** |
| 7  5  8  7 | O número correto é maior.  O número correto é menor.  Parabéns! Você acertou. |
| 5  4  5 | O número correto é maior.  Parabéns! Você acertou. |
| -2  -1  -3  -2 | O número correto é menor.  O número correto é maior.  Parabéns! Você acertou. |

1. Faça um programa que peça ao usuário para digitar uma sequência de inteiros. O programa deve parar quando **0** for digitado, que será desconsiderado na sequência de números lidos. No final, você deve apresentar a quantidade de números lidos, o maior inteiro e a média aritmética simples dos inteiros.

**Entrada**

A entrada consistirá de uma sequência de inteiros que será terminada quando o valor 0 for digitado, o qual não fará parte da sequência. É possível que a sequência não tenha nenhum número (nesse caso considere 0 como o maior número da sequência).

**Saída**

Apresente ***x***, ***y*** e ***z***, um por linha, onde ***x***, ***y*** e ***z*** representam, respectivamente, a quantidade de números, o maior número e a média dos inteiros da sequência com 2 casas decimais após a vírgula.

|  |  |
| --- | --- |
| **Exemplo de Entrada** | **Exemplo de Saída** |
| 3  3  -1  -2  -4  -5  0 | 6  3  -1.00 |
| -1  -2  -3  -4  -5  0 | 5  -1  -3.00 |
| 2  2  -2  -2  0 | 4  2  0.00 |

1. Leia dois inteiros do teclado e apresente na tela o fatorial de cada um, na mesma linha, usando laço de repetição.

**Entrada**

Dois inteiros maiores ou iguais a zero.

**Saída**

Calcule e imprima o fatorial de cada um, separados por espaço.

|  |  |
| --- | --- |
| **Exemplo de Entrada** | **Exemplo de Saída** |
| 2  2 | 2 2 |
| 4  3 | 24 6 |
| 0  1 | 1 1 |

1. Escreva um programa que leia dois valores inteiros ***X*** e ***Y***. A seguir, mostre uma sequência de 1 até ***Y***, passando para a próxima linha a cada ***X*** números.

**Entrada**

A entrada contém duas linhas. A primeira linha será o ***X*** e a segunda o ***Y***, onde ***X, Y > 0***.

**Saída**

Cada sequência deve ser impressa em uma linha apenas, com 1 espaço em branco entre cada número, conforme exemplo abaixo. **Não deve haver espaço em branco após o último valor de cada linha.**

**Nota:** Para que o fim da linha não seja impresso após o uso do comando ***print***, é necessário alterar o argumento ***end***. Por padrão, o argumento *end* é: *end= ‘\n’,* que é o caractere que representa o final da linha. No exemplo abaixo o argumento *end= “ “* do primeiro comando *print* suprime a nova linha no final da saída, que é o motivo pelo qual as duas saídas vão aparecer na mesma linha separadas por espaço. Nesse exemplo a saída será: ***5 laranjas***.

**a = 5**

**print(a, end= “ “)**

**a = “laranjas”**

**print(a)**

|  |  |
| --- | --- |
| **Exemplo de Entrada** | **Exemplo de Saída** |
| 3  99 | 1 2 3  4 5 6  7 8 9  10 11 12  ...  97 98 99 |
| 2  9 | 1 2  3 4  5 6  7 8  9 |
| 4  3 | 1 2 3 |

1. Faça um algoritmo para ler um valor ***T***, um valor ***A*** e um valor ***N***. Leia ***T*** vezes valores ***A*** e ***N*** e imprima a soma dos ***N*** números a partir de ***A*** (inclusive), para cada um dos valores ***A*** e ***N*** lidos. Imprima também cada um dos ***N*** números a partir de ***A***, incluindo o ***A***.

**Entrada**

A entrada contém somente valores inteiros, sendo ***T ≥ 0*** e ***N > 0***. Na primeira linha será lido o valor ***T*** e nas próximas ***T*** linhas serão lidos os valores de ***A*** e ***N***, separados por espaço.

**Saída**

Escreva na tela, para cada dupla de ***A*** e ***N*** lidos, cada um dos ***N*** números a partir de ***A***, separados por espaço. Logo em seguida imprima ***X***, onde ***X*** representa a soma dos ***N*** números a partir de ***A***, conforme exemplo de saída. **Não deve haver espaços em branco após o último valor de cada linha.**

**Nota:** Novamente, lembre-se que para ler vários valores em uma mesma linha, usamos ***input().split()***. Se o argumento de split for vazio, o separador das variáveis é um espaço em branco. Porém, lembre-se que input lê apenas strings do teclado, portanto você deverá converter as strings. No exemplo a seguir, o usuário digita valores separados por um espaço em branco e aperta enter para enviá-los, então, o programa lê esses valores separados por espaços como strings (na ordem em que aparecem), guardados nas variáveis correspondentes e os converte para floats. Depois imprime os valores, **um por linha**:

***A, B = input().split()***

***A, B = [float(A), float(B)]***

***print(“%.0f\n%.0f”%(a,b))***

|  |  |
| --- | --- |
| **Exemplo de Entrada** | **Exemplo de Saída** |
| 3  1 3  4 5  0 10 | 1 2 3  6  4 5 6 7 8  30  0 1 2 3 4 5 6 7 8 9  45 |
| 2  5 5  4 3 | 5 6 7 8 9  35  4 5 6  15 |
| 3  -1 4  -5 10  -3 1 | -1 0 1 2  2  -5 -4 -3 -2 -1 0 1 2 3 4  -5  -3  -3 |

1. Leia um valor inteiro ***N*** que é a quantidade de casos de teste que vem a seguir. Cada caso de teste consiste de dois inteiros ***X*** e ***Y***. Você deve apresentar a soma de ***Y*** ímpares consecutivos a partir de ***X*** inclusive o próprio ***X*** se ele for ímpar. Por exemplo: para a entrada 4 5, a saída deve ser 45, que é equivalente à: 5 + 7 + 9 + 11 + 13, para a entrada 7 4, a saída deve ser 40, que é equivalente à: 7 + 9 + 11 + 13. No final imprima também a maior e a menor soma.

**Entrada**

A primeira linha de entrada é um inteiro ***N*** ***> 0*** que é a quantidade de casos de teste que vem a seguir. Cada caso de teste consiste em uma linha contendo dois inteiros ***X*** e ***Y***, onde ***Y > 0***.

**Saída**

Imprima a soma ***S*** dos ***Y*** consecutivos números ímpares a partir do valor ***X***, para cada ***X*** e ***Y*** lidos. Imprima também a maior e a menor soma ***S***, conforme exemplo abaixo.

|  |  |
| --- | --- |
| **Exemplo de Entrada** | **Exemplo de Saída** |
| 4  -2 5  3 3  -10 3  4 4 | 15  15  -21  32  32  -21 |
| 3  -5 1  -3 2  -10 3 | -5  -4  -21  -4  -21 |
| 2  -5 2  -5 4 | -8  -8  -8  -8 |

1. Raphael e Renata estão cursando Teoria dos Números juntos, no departamento de matemática da Universidade de Brasília. Eles se deparam com a seguinte hipótese: “Para todo inteiro positivo ***n*** e ***m*** temos que é um número primo”. Porém, eles percebem que essa hipótese é falsa, pois a Renata rapidamente nota que basta usar , assim:

que não é primo.

De modo que para ***n > 2***, ***m*** pode ser **n – 2**. Se ***n = 7***, por exemplo, então 7 5 + 1 = 36, que não é primo**.** Se ***n ≤ 2***, podemos usar ***m = n + 2***.

Entretanto, Raphael gosta da Renata e quer impressioná-la, apresentando não apenas qualquer contra-exemplo, mas sim o menor ***m*** tal que não seja primo (para ***n = 7***, também poderíamos usar ***m = 1***). Você pode escrever um programa para ajudá-lo, dado o inteiro ***n***?

**Entrada**

A entrada consistirá apenas de um inteiro ***n*** (1 ≤ ***n*** ≤ 1000) – o ***n*** da hipótese.

**Saída**

Imprima na tela o menor ***m ≥ 1*** tal que não seja um número primo. É garantido que esse ***m*** existe.

**Dica:** Use dois loops, um para o ***m*,** que enquanto ***m*** for menor ou igual a 1000 testará se é primo.

**Nota**

Para o primeiro exemplo, 3∙1 + 1 = 4, a saída será 1.

Para o segundo exemplo, 4∙1 + 1 = 5, nós não podemos imprimir 1 porque 5 é primo. Porém, *m = 2* está tudo bem, visto que 4∙2 + 1 = 9, que não é primo.

Para o terceiro exemplo, 10∙2 + 1 = 21, imprimimos 2.

|  |  |
| --- | --- |
| **Exemplo de Entrada** | **Exemplo de Saída** |
| 3 | 1 |
| 4 | 2 |
| 10 | 2 |