Nome: Matheus de Amorim Favero – Ciência da Computação

Algoritmos – Professor Patrick

Atividade de Reforço – 28/04

1. Com relação às estruturas de repetição faça o seguinte:
   1. Dê exemplos de estruturas de repetição.

R. Algumas estruturas de repetição são “para” (for) e “enquanto” (while).

* 1. Qual a função destas estruturas?

R. A função destas estruturas de repetição é fazer com que determinada instrução seja executada repetidas vezes dentro de uma determinada condição que será definida como parâmetro para a estrutura.

* 1. Qual a relação entre estruturas de repetição e legibilidade de um código?

R. As estruturas de repetição reduzem a quantidade de linhas de um código pois determinam instruções a serem executadas dentro de determinadas vezes. Desta forma, com as estruturas, fazemos mais ações, utilizando menos linhas de código; facilitando sua leitura e a identificação de cada função.

* 1. Dê exemplo de um programa que não utilize uma estrutura de repetição mas execute a mesma tarefa 3 vezes seguidas. Depois, transforme esse programa para uma versão que faça a mesma coisa que o programa anterior, mas usando uma estrutura de repetição.

R. Definição: **Um algoritmo que receba um número dado pelo usuário e o mostre multiplicado por 1, 2 e 3.**

Sem estrutura de repetição:

num = float(input("Digite um número: "))

print("{0} x 1 = {1}".format(num, num\*1))

print("{0} x 2 = {1}".format(num, num\*2))

print("{0} x 3 = {1}".format(num, num\*3))

Utilizando estrutura de repetição:

num = float(input("Digite um número: "))

for i in range(3):

print("%d \* %d = %d" % (num, i+1, num\*(i+1)))

* 1. Numere as linhas dos programas desenvolvidos por você e mostre em que ordem elas são executadas (desde o início até o fim da execução do programa).

R. Sem Estrutura

|  |  |
| --- | --- |
| Início | Início |
| Linha 1 | Recebe o valor digitado pelo usuário e armazena na variável “num” como um número decimal. |
| Linha 2 | Realiza a impressão de uma mensagem e interpolado nesta mensagem há dois valores variáveis; o **primeiro** é o valor digitado pelo usuário (num) e o **segundo** é o número digitado multiplicado pelo número 1. |
| Linha 3 | Realiza outra impressão com dois valores variáveis interpolados na mensagem; o **primeiro** é a variável “num”, e o **segundo** é a variável “num” multiplicada por 2. |
| Linha 4 | Realiza a última impressão com dois valores variáveis interpolados na mensagem; o **primeiro** é a variável “num”, e o **segundo** é a variável “num” multiplicada por 3. |
| Fim | Fim |

Com Estrutura:

|  |  |
| --- | --- |
| Linha 1 | Recebe o valor digitado pelo usuário e armazena na variável “num” como um número decimal. |
| Linha 2 | Define a condição de laço: para cada valor *(for i)* até a distância 3 *(in range(3))* faça*(:)* |
| Linha 3 | Define ação a ser executada dentro do laço: Imprime uma mensagem, nesta mensagem há uma string (texto) e três valores variáveis interpolados no meio da mensagem e formatados utilizando-se a função .format(). O primeiro valor é a variável “num”; o segundo valor é o estado do valor “i” somado com 1, tendo por objetivo, alterar o número da regra de repetição do laço e o número visível na mensagem do usuário; o terceiro valor é a variável “num” multiplicada pela regra de repetição do laço somada com 1, pois a adição nesta regra determina o próximo multiplicador da variável. Isso estabelece quem multiplica o valor de “num” e monta o padrão de uma tabuada. |

* 1. A utilização de estruturas de repetição aumenta o diminuem o número de linhas necessárias para se executar instruções que precisam ser repetidas? Por que?

R. Sim, diminui, pois com as estruturas de repetição, é possível definir uma condição para que determinada instrução seja executada até que a condição inicial se prove FALSA.

1. Explique como funciona cada comando seguinte (“se...senão”, “enquanto...”, “para...”), sob quais condições eles são ou não executados? Implemente um programa em que os três comandos são usados juntos.

R. O comando “se... senão” é dividido em duas sessões; o “se...” é executado somente se determinada condição se prove VERDADEIRA. Caso a condição se prove FALSA ou diferente de verdadeira, o bloco de código do comando “senão” será executado.

O comando “enquanto” é uma estrutura de repetição que executará determinada instrução **até que** a condição inicial se prove FALSA.

O comando “para” é uma estrutura de repetição que executará determinada ação com base em uma instrução dentro de um limite/distância já definido de forma prévia. O laço ocorrerá até que instrução seja provada como falsa.

1. Sobre as diferentes formas de representação de algoritmos responda:
   1. De quais formas os algoritmos são representados por meio dos fluxogramas?

R. O fluxograma apresenta uma representação gráfica de um algoritmo lógico, cujo se define mais sucinta que a representação textual. Deste modo, há regras de simbologia que devem ser seguidas para representar um fluxo de código por meio de um fluxograma.

* 1. Quais os problemas da representação de algoritmos por meio da descrição narrativa?

R. Devido a ambiguidade presente na linguagem natural, um computador não poderia interpretar um algoritmo neste formato, da mesma forma que um humano.

* 1. Quais as vantagens de usar o pseudocódigo como forma de representação de algoritmos? O pseudocódigo é uma linguagem de programação? Ele está vinculado a alguma linguagem específica?

R. A vantagem é que sua transcrição para qualquer linguagem é quase direta, podendo ser reutilizado em todo cenário. Pseudocódigo não é uma linguagem de programação, mas sim uma forma de se descrever o processo de lógica por trás de um algoritmo. Não está vinculado a nenhuma linguagem específica.

1. Implemente o algoritmo seguinte em Python. Preencha a última coluna da tabela seguinte, indicando quantas vezes o programa imprime a frase especificada no código, considerando as diferentes condições indicadas:

n🡨0

enquanto(<condição>):

escreva('REVISANDO CONCEITOS')

n🡨n+1

|  |  |
| --- | --- |
| CONDIÇÕES  DIFERENTES A  SEREM TESTADAS | NÚMERO DE  IMPRESSÕES DA  FRASE |
| n<>10 | 10 |
| n<=10 | 11 |
| n<9 | 10 |
| n=10 | 0 |

1. Implemente o algoritmo seguinte em Python. Preencha a última coluna da tabela seguinte, indicando qual a saída (sequência de números impressa), ao preencher as linhas 1 e 2 com os valores indicado:

Linha 1 ??????

Linha 2 enquanto(<condição>):

Linha 3 escreva(n)

Linha 4 n🡨 n+2

|  |  |
| --- | --- |
| VALORES PARA  LINHA 1 E LINHA 2 | O QUE É IMPRESSO NA  TELA? |
| linha 1: n🡨1  linha 2: (n>21) | Nada |
| linha 1: n🡨1  linha 2: (n<>21) | 1 3 5 7 9 11 13 15 17 19 |
| linha 1: n🡨0  linha 2: (n<=21) | 0 2 4 6 8 10 12 14 16 18 20 |
| linha 1: n🡨0  linha 2: (n>21) | Nada |

1. Implemente o algoritmo seguinte em Python. Considerando que objetivo do algoritmo é imprimir apenas a soma dos números negativos dentre os números informados pelo usuário, preencha a tabela seguinte mostrando qual a saída obtida pelo usuário para as diferentes condições indicadas e supondo que ele digite a seguinte sequência de números: [-1, 0, -0.5, 2, -2].

|  |  |
| --- | --- |
| SE A CONDIÇÃO FOR... | QUAL A SAÍDA OBTIDA? |
| numero<=1 | O número digitado não entrou na soma (3 vezes)  A soma dos números negativos foi: 1 |
| numero>0 | O número digitado não entrou na soma  A soma dos números negativos foi: 10 |
| numero<= -1 | O número digitado não entrou na soma (5 vezes)  A soma dos números negativos foi: 0 |
| numero<0 | O número digitado não entrou na soma (5 vezes)  A soma dos números negativos foi: 0 |

1. Faça o que se pede:
   1. Implemente o algoritmo seguinte em Python e mostre qual a saída do programa:

R. Imprimindo o número 1

Imprimindo o número 2

Imprimindo o número 3

Imprimindo o número 4

Imprimindo o número 5

Imprimindo o número 6

Imprimindo o número 7

Imprimindo o número 8

Imprimindo o número 9

Imprimindo o número 10

Imprimindo o número 11

Imprimindo o número 12

Imprimindo o número 13

* 1. Agora faça 4 diferentes versões do programa em Python que utilize o comando “for” no lugar de “while”. Para cada uma das versões você deve implementar o “for” (para) considerando o que é indicado na tabela abaixo. Para cada uma das versões mostre a saída do programa:

|  |  |
| --- | --- |
| COMO DEVE SER IMPLEMENTADO O PARA... | QUAL A SAÍDA OBTIDA? |
| para x de 1 até 14:  escreva('Imprimindo o número ', x) | Imprimindo o número 0  Imprimindo o número 1  Imprimindo o número 2  Imprimindo o número 3  Imprimindo o número 4  Imprimindo o número 5  Imprimindo o número 6  Imprimindo o número 7  Imprimindo o número 8  Imprimindo o número 9  Imprimindo o número 10  Imprimindo o número 11  Imprimindo o número 12  Imprimindo o número 13 |
| para x de 0 até 13:  escreva('Imprimindo o número ', x) | Imprimindo o número 0  Imprimindo o número 1  Imprimindo o número 2  Imprimindo o número 3  Imprimindo o número 4  Imprimindo o número 5  Imprimindo o número 6  Imprimindo o número 7  Imprimindo o número 8  Imprimindo o número 9  Imprimindo o número 10  Imprimindo o número 11  Imprimindo o número 12 |
| para x de 1 até 13:  escreva('Imprimindo o número ', x) | Imprimindo o número 0  Imprimindo o número 1  Imprimindo o número 2  Imprimindo o número 3  Imprimindo o número 4  Imprimindo o número 5  Imprimindo o número 6  Imprimindo o número 7  Imprimindo o número 8  Imprimindo o número 9  Imprimindo o número 10  Imprimindo o número 11  Imprimindo o número 12 |
| para x de 1 até 12:  escreva('Imprimindo o número ', x) | Imprimindo o número 0  Imprimindo o número 1  Imprimindo o número 2  Imprimindo o número 3  Imprimindo o número 4  Imprimindo o número 5  Imprimindo o número 6  Imprimindo o número 7  Imprimindo o número 8  Imprimindo o número 9  Imprimindo o número 10  Imprimindo o número 11 |

1. Implemente o algoritmo seguinte em python. Indique os valores iniciais das variáveis A e B e seus valores finais quando o programa termina. Comente sobre o seguinte: a segunda instrução de atribuição de um valor para A, alterou o valor da variável B?