

EXPLICAÇÃO DAS QUESTÕES DO SIMULADO.

AO FINAL DO SIMULADO VOCÊ TAMBÉM PODE CLICAR EM REVISAR QUESTÕES PARA TER ACESSO A ESTE CONTEÚDO

Pergunta 1: **Correto**

Marque abaixo uma ou mais alternativas que considerar corretas

O Comando de entrada de dados em que você atribui o valor digitado pelo usuário a uma variável é o:

☐ `std::cout <<`

☐ `std::cin <<`

☒ `std::cin >>`

(Correto)

☒ `cin >>` desde que você use o comando `using namespace std;`

(Correto)

Explicação

O correto são as alternativas `std::cin >>` e `cin` se for usado `using namespace std;`

Exemplo de código:

```
#include <iostream>

int main()
{
    float Num1;
    std::cout << "Digite o Numero: ";
    std::cin >> Num1;
    return 0;
}
```

Ou utilizando `using namespace std;`

```
#include <iostream>

using namespace std;

int main()
{
```

```
float Num1;

cout << "Digite o Numero: ";

cin >> Num1;

return 0;

}
```

Pergunta 2: **Correto**
Analise código a seguir:

Imagem maior

```
#include <iostream>

int main()
{
    float Numero01, Numero02;
    int Resultado;
    std::cout << "Digite o primeiro numero: ";
    std::cin >> Numero01;
    std::cout << "Numero01 = " << Numero01 << std::endl;
    std::cout << "Digite o segundo numero: ";
    std::cin >> Numero02;
    std::cout << "Numero02 = " << Numero02 << "\n";
    Resultado = Numero01 % Numero02;
    std::cout << "A Resto da Divisao inteira dos dois Numeros: " << Resultado << "\n";
    system("PAUSE");
}
```

```
1. #include <iostream>
2.
3. int main()
4. {
5.     float Numero01, Numero02;
6.     int Resultado;
7.     std::cout << "Digite o primeiro numero: ";
8.     std::cin >> Numero01;
9.     std::cout << "Numero01 = " << Numero01 << std::endl;
10.    std::cout << "Digite o segundo numero: ";
11.    std::cin >> Numero02;
12.    std::cout << "Numero02 = " << Numero02 << "\n";
13.    Resultado = Numero01 % Numero02;
14.    std::cout << "A Resto da Divisao inteira dos dois Numeros: " << Resultado << "\n";
15.    system("PAUSE");
16. }
```

O código acima irá compilar sem erros.

Esta afirmação está:



CORRETA



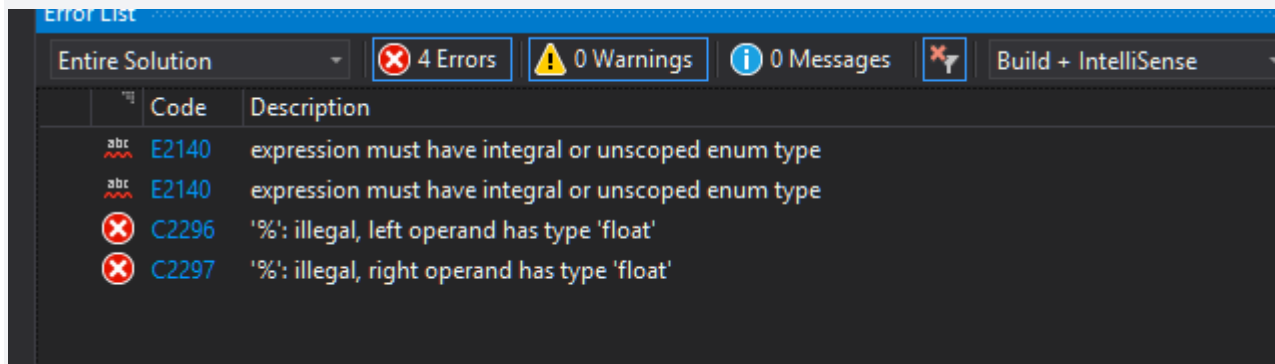
INCORRETA

(Correto)

Explicação

Está INCORRETA pois o Operador % Resto da Divisão Inteiro ou Módulo só aceita números inteiros ou int para realizar o cálculo. O código apresentado na questão os números são float e deveriam ser do tipo int.

O erro de compilação vai aparecer:



Assim o código correto é este

```
#include <iostream>

int main()
{
    int Numero01, Numero02;
    int Resultado;
    std::cout << "Digite o primeiro numero: ";
    std::cin >> Numero01;
    std::cout << "Numero01 = " << Numero01 << std::endl;
    std::cout << "Digite o segundo numero: ";
    std::cin >> Numero02;
    std::cout << "Numero02 = " << Numero02 << "\n";
    Resultado = Numero01 % Numero02;
    std::cout << "A Resto da Divisao inteira dos dois Numeros: " << Resultado << "\n";
    system("PAUSE");
}
```

Pergunta 3: **Correto**

Marque a seguir a única resposta correta:

Em relação a precedência de operador analise a expressão abaixo:

[Imagem maior](#)

Result = N1 + N2 * N3 - N4 * N5;

Esta mesma expressão acima pode ser reescrita usando parêntesis da seguinte forma:

Result = (N1 + N2) * ((N3 - N4) * N5);



Result = N1 + (N2 * N3) - (N4 * N5);

(Correto)



Result = ((N1 + (N2 * N3)) - N4) * N5;



Result = (N1 + N2 * (N3 - N4) * N5);

Explicação

A forma correta é: **Result = N1 + (N2 * N3) - (N4 * N5);** pois primeiro na ordem de precedência ocorre a multiplicação

you can also test with the code below

```
#include <iostream>

int main()
{
    int Result;

    int N1{ 1 }, N2{ 2 }, N3{ 3 }, N4{ 4 }, N5{ 5 };

    Result = N1 + (N2 * N3) - (N4 * N5);

    std::cout << Result << "\n";

    system("PAUSE");
}
```

Pergunta 4: **Correto**

Considere o código abaixo

```
1. #include <iostream>
2. int main()
3. {
4.     //Se Numero01 for igual a Numero02 o resultado será 1 para verdade(true) ou 0 para false
5.     int Numero01 = 10, Numero02 = 20;
6.     bool Comparacao;
7.     Comparacao = (Numero01 = Numero02);
8.     std::cout << "Numero01 igual a Numero02?: " << Comparacao << "\n";
9.     system("PAUSE");
10.    return 0;
11. }
```

Comparação receberá o valor lógico da comparação de Numero01 e Numero02 que será 1 para true ou 0 para false

Como os números Numero01 e Numero02 são diferentes o resultado da operação será zero e irá aparecer na tela 0

A afirmação acima está:

CORRETA

INCORRETA

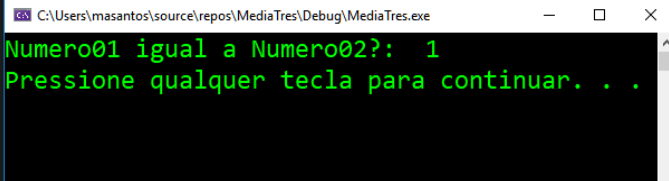
(Correto)

Explicação

É incorreta pois o erro que aconteceu é que o operador de igualdade não é = e sim ==. Este é um erro clássico que comentei na aula.. Veja que embora os números sejam diferentes pois 10 é diferente de 20 o resultado deu que eles são iguais 1. Deveria ter dado zero 0.

```
#include <iostream>

int main()
{
    //Se Numero01 for igual a Numero02 o resultado será 1 para verdade(true) ou 0 para false
    int Numero01 = 10, Numero02 = 20;
    bool Comparacao;
    Comparacao = (Numero01 = Numero02);
    std::cout << "Numero01 igual a Numero02?: " << Comparacao << "\n";
    system("PAUSE");
    return 0;
}
```



Observe que `Comparacao = (Numero01 = Numero02);` faz com que primeiro Numero01 receba o valor de Numero02 pois está sendo usado o operador de atribuição = ao invés do operador de igualdade ==

O correto seria: `Comparacao = (Numero01 == Numero02);`

Com o código correto ai sim o resultado fica ok

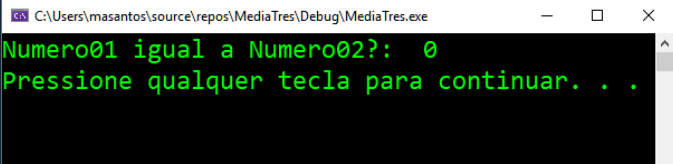
```
#include <iostream>

int main()
{
    //Se Numero01 for igual a Numero02 o resultado será 1 para verdade(true) ou 0 para false

    int Numero01 = 10, Numero02 = 20;
    bool Comparacao;
    Comparacao = (Numero01 == Numero02);
    std::cout << "Numero01 igual a Numero02?: " << Comparacao << "\n";
    system("PAUSE");
    return 0;
}
```

```
#include <iostream>

int main()
{
    //Se Numero01 for igual a Numero02 o resultado será 1 para verdade(true) ou 0 para false
    int Numero01 = 10, Numero02 = 20;
    bool Comparacao;
    Comparacao = (Numero01 == Numero02);
    std::cout << "Numero01 igual a Numero02?: " << Comparacao << "\n";
    system("PAUSE");
    return 0;
}
```



Pergunta 5: **Correto**

Marque uma ou mais opções de resposta a seguir...

Em relação as informações a abaixo é correto afirmar que:



A **lógica** é a organização do pensamento humano, ato de pensar, fazemos uma seleção de ideias procurando associá-las da melhor maneira possível em função de uma conclusão. É o encadeamento coerente de alguma coisa que obedece a certas regras e convenções.

(Correto)



Um **algoritmo** é uma sequência infinita de instruções bem definidas e não ambíguas, onde cada uma das quais pode ser executada mecanicamente dentro de um determinado período finito de tempo e esforço



Um **pseudocódigo** é uma forma genérica de escrever um algoritmo, utilizando uma linguagem simples (nativa a quem o escreve, de forma a ser entendida por qualquer pessoa) sem necessidade de conhecer a sintaxe de nenhuma linguagem de programação. Um exemplo de pseudocódigo é o Portugol, que é utilizada no compilador Visualg

(Correto)



A **entrada (input)**. É simplesmente o resultado de todo o processamento, em todo processamento temos dados gerados como resultado, essas entradas, podem ser impressas na tela, em papel, armazenadas em um arquivo, ou até mesmo servir como entrada para um outro processo.

Explicação

A afirmação acerca de algoritmo está incorreta pois ele precisa acabar e ter um fim e portanto, precisa ser finito

Um **algoritmo** é uma sequência **FINITA** (e não infinita) de instruções bem definidas e não ambíguas, onde cada uma das quais pode ser executada mecanicamente dentro de um determinado período finito de tempo e esforço

Já a afirmação acerca de entrada(Input) está incorreta pois, na realidade o texto está se referenciando a saída(output)

O correto seria: A **Saída (output)**. É simplesmente o resultado de todo o processamento, em todo processamento temos dados gerados como resultado, essas **Saídas**, podem ser impressas na tela, em papel, armazenadas em um arquivo, ou até mesmo servir como entrada para um outro processo.

Pergunta 6: **Correto**

Em relação as afirmações abaixo marque uma ou mais opções que achar verdadeira

☐

A **Conjunção(E)** AND, que em C++ é representa por **&&**, somente retornará VERDADEIRO: SE TODAS AS CONDIÇÕES FOREM FALSAS!

☒

Uma tabela que representa todas as possibilidades de valores lógicos é denominada tabela-verdade.

(Correto)

☒

A **Disjunção(ou)** que em C++ é representa por **||**, somente retornará FALSO: SE TODAS AS CONDIÇÕES FOREM FALSAS!

(Correto)

☒

Operador Lógico **NÃO, NOT, !** retorna o oposto do valor lógico da expressão avaliada.
Não false = True
e Não True = False.

(Correto)

Explicação

A **Conjunção(E)** AND, que em C++ é representa por **&&**, somente retornará VERDADEIRO: SE TODAS AS CONDIÇÕES FOREM VERDADEIRAS E NÃO FALSAS COMO INCORRETAMENTE ASSINALADO NA QUESTÃO

Pergunta 7: **Correto**

Para utilizar os especificadores de formato(%d, %f, %lf, %c, %s) que irão compor o texto de saída de acordo com a variável a qual cada especificador está relacionado, podemos usar o comando std::printf. Sendo que printf vem de "print formatted" (imprimir formatado)

A afirmação acima está:

☒

CORRETA

(Correto)

☐

INCORRETA

Explicação

Isso mesmo está correta!

Lembre da aula:

%d Número decimal inteiro (int).

%f Número decimal de ponto flutuante. serve tanto para float como para double.

%lf Serve para double. (l-long , f-float - lf)

%c Caractere: imprime o caractere que tem o código ASCII correspondente ao valor dado.

%s Sequência de caracteres (string, em inglês).

obs: se quiser colocar um sinal de porcentagem use %% Imprime um %

Pergunta 8: **Correto**

Analise o código abaixo e responda:

[Imagem maior](#)

```
#include <iostream>

int main()
{
    //Se Numero01 for igual a Numero02 o resultado será 1 para verdade(true) ou 0 para false
    int Num1 = 10, Num2 = 20, Num3, Num4;
    Num3 = Num1++;
    Num4 = ++Num2;
    std::cout << "Num1 = " << Num1 << std::endl;
    std::cout << "Num2 = " << Num2 << "\n";
    std::cout << "Num3 = " << Num3 << std::endl;
    std::cout << "Num4 = " << Num4 << "\n";
    system("PAUSE");
    return 0;
}
```

Após ser executado este código os valores de Num1, Num2, Num3 e Num4 serão respectivamente:

☐

Num1 = 11 Num2 = 21 Num3 = 11 e Num4 = 20

☐

Num1 = 11 Num2 = 21 Num3 = 11 e Num4 = 21

☒

Num1 = 11 Num2 = 21 Num3 = 10 e Num4 = 21

(Correto)

☐

Num1 = 10 Num2 = 21 Num3 = 11 e Num4 = 21

Num1 = 10 Num2 = 20 Num3 = 10 e Num4 = 20

Explicação

Verifique o código:

```
#include <iostream>

int main()
{
    //Se Numero01 for igual a Numero02 o resultado será 1 para verdade(true) ou 0 para false
    int Num1 = 10, Num2 = 20, Num3, Num4;

    Num3 = Num1++;
    Num4 = ++Num2;

    std::cout << "Num1 = " << Num1 << std::endl;
    std::cout << "Num2 = " << Num2 << "\n";
    std::cout << "Num3 = " << Num3 << std::endl;
    std::cout << "Num4 = " << Num4 << "\n";

    system("PAUSE");

    return 0;
}
```

Quando o código chega no comando **Num3 = Num1++**

temos um comando de atribuição. Pela aula você lembra que os operadores ++ e -- podem estar pré-fixados ou pós-fixados

No caso de **Num3 = Num1++** o operador ++ está pós-fixado e isso indica que primeiro o compilador deve efetuar a atribuição e só depois deverá realizar o incremento de Num1

Logo é como se tivéssemos: **Num3 = Num1** e neste caso Num3 passa valer 10. E depois (pós) é que é realizado o incremento de Num1 e assim Num1++ a variável Num1 passa a ter o valor 11;

Já na linha **Num4 = ++Num2**; temos que o operador de incremento ++ está pré-fixado e desta forma estamos indicando ao compilador para primeiro(pré) ele deve incrementar a variável Num2 para só depois atribuir este valor a Num4

Assim a variável Num2 para a valer: $20 + 1 = 21$; e Num4 recebe este valor 21 de Num2

Portanto a resposta correta é: **Num1 = 11 Num2 = 21 Num3 = 10 e Num4 = 21**

```
#include <iostream>
```

```
int main()
```

```
{
```

```
    //Se Numero01 for igual a Numero02 o resultado será 1 para ver
```

```
    int Num1 = 10, Num2 = 20, Num3, Num4;
```

```
    Num3 = Num1++;
```

```
    Num4 = ++Num2;
```

```
    std::cout << "Num1 = " << Num1 << std::endl;
```

```
    std::cout << "Num2 = " << Num2 << "\n";
```

```
    std::cout << "Num3 = " << Num3 << std::endl;
```

```
    std::cout << "Num4 = " << Num4 << "\n";
```

```
    system("PAUSE");
```

```
    return 0;
```

```
}
```

C:\Users\masantos\source\repos\MediaTres\Debug\MediaTres.exe

Num1 = 11

Num2 = 21

Num3 = 10

Num4 = 21

Pressione qualquer tecla para continuar.