

<b>CURSO:</b> Segurança da Informação		<b>DISCIPLINA:</b> Programação I		<b>Turma</b> SEG NA1	
<b>DOCENTE:</b> Alan Henrique Pardo de Carvalho					
<b>AVALIAÇÃO OFICIAL:</b> <input checked="" type="checkbox"/> N1 <input type="checkbox"/> N2 <input type="checkbox"/> N3				<b>DATA:</b> 22/04/2023	
<b>NOMES DOS ALUNOS:</b>		Anselmo Antunes Ribeiro			
<p><b>Condições:</b> Consulta aos materiais é permitida. Essa prova pode ser feita individualmente ou em duplas. Se for feita em duplas, apenas um integrante da dupla precisa entregar a prova no Teams, mas os NOMES COMPLETOS dos dois alunos da dupla precisarão estar no campo acima. A compreensão das questões faz parte da prova. Nas questões onde seja necessário marcar uma alternativa, basta clicar na <i>checkbox</i> correspondente para ativá-la ou desativá-la. Salve esse arquivo preenchido em formato PDF e entregue o PDF junto com os códigos fonte dos programas na tarefa do Teams relativa à prova. Você pode usar qualquer ambiente para escrever os programas, mas tenha em mente que eles serão testados e validados no Dev C++.</p> <p><b>Horário máximo para entrega:</b> até as 23h59 do dia 22/04/2023. O Teams fechará automaticamente a tarefa e não haverá como entregar depois.</p>					

1. Marque a caixa correspondente a “SIM” ou “NÃO” (só pode ser uma das alternativas) para cada afirmação a seguir. Cada afirmação com a marcação correta vale 0,05 ponto.

AFIRMAÇÃO	SIM	NÃO
Programar em Assembly não exige um conhecimento avançado da arquitetura de hardware e software do sistema no qual se deseja programar.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Uma linguagem de programação de alto nível facilita o trabalho do programador por estar mais próxima dos detalhes de arquitetura de hardware e software do sistema no qual se deseja programar.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Uma instrução em uma linguagem de alto nível, como Python, representa diversas instruções em uma linguagem de baixo nível, como Assembly.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Um programa que pede dois números ao usuário, calcula e apresenta o valor da soma desses dois números está utilizando a estrutura sequencial.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Um programa que pede dois números ao usuário e verifica se o quadrado deles é par está utilizando a estrutura condicional.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Um programa que solicita ao usuário que informe, para cada produto adquirido, a quantidade de unidades e o preço unitário, está utilizando a estrutura de repetição ( <i>loop</i> ).	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Um programa compilado, para ser executado em um outro computador, necessita que seja enviado o código fonte para a pessoa que irá executar o programa.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Um programa interpretado tem o código objeto gerado pelo interpretador sempre que o programa for executado.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Assembly é uma linguagem altamente portátil pois possui alto desempenho, relacionado diretamente ao processador do computador onde o programa será executado.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
É possível que um único programa utilize a estrutura sequencial, condicional e de repetição ( <i>loop</i> ).	<input checked="" type="checkbox"/>	<input type="checkbox"/>

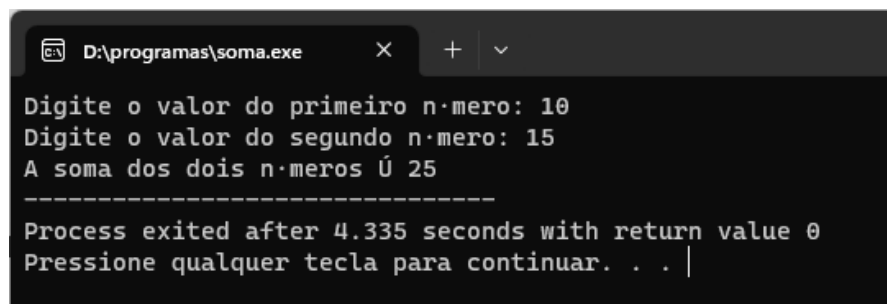
2. Responda no espaço indicado após o programa por qual motivo a linha 3 é necessária. (0,5 ponto)

```

1  /* olamundo.cpp - exibe a mensagem "Olá, mundo" na tela */
2
3  #include <stdio.h>
4  #include <locale.h>
5
6  int main() {
7      setlocale(LC_ALL, "");
8      printf("Olá, mundo");
9      return 0;
10 }
```

O uso da biblioteca é necessário, pois ela permite o uso das funções de entrada e saída de dados.  
Ex: scanf, printf.

3. Responda no espaço indicado após o programa por qual motivo o texto apareceu com caracteres especiais no lugar das letras acentuadas como no exemplo abaixo. (0,5 ponto)



```
D:\programas\soma.exe
Digite o valor do primeiro n-mero: 10
Digite o valor do segundo n-mero: 15
A soma dos dois n-meros ũ 25
-----
Process exited after 4.335 seconds with return value 0
Pressione qualquer tecla para continuar. . . |
```

```
1  /* soma.cpp - exibe a soma de dois números na tela
2     por Alan Carvalho - 22/04/2023
3  */
4
5  #include <stdio.h>
6  #include <locale.h>
7
8  int main() {
9      int num1, num2;
10     printf("Digite o valor do primeiro número: ");
11     scanf("%d", &num1);
12     printf("Digite o valor do segundo número: ");
13     scanf("%d", &num2);
14     printf("A soma dos dois números é %d", num1 + num2);
15     return 0;
16 }
17
```

O texto apareceu com caracteres especiais, pois mesmo a biblioteca " locale.h " ter sido inserida, no código não foi usada a função " setlocale(LC\_ALL," ") " .

4. Qual a relação entre os tipos de dados em uma linguagem de programação e o espaço de memória RAM ocupado pelos dados de um programa? (0,5 ponto)

Estão diretamente relacionados, pois cada tipo de dado possui diferentes tamanhos em bytes e esse tamanho determina a quantidade de espaço da memória que será utilizada.  
Portanto, se um programa utiliza muitos dados de um tipo específico, o tamanho total ocupado na memória RAM pode ser significativamente maior, podendo causar sobrecarga de memória.

5. O que significa dizer que os nomes de variáveis em C são *case sensitive*? Dê um exemplo. (0,5 ponto)

Significa que a linguagem faz diferenciação entre letras maiúsculas e minúsculas. Por exemplo, uma variável "idade" é diferente da variável "Idade" ou "IDADE".

6. Quais dos seguintes identificadores são inválidos na linguagem C? Explique o motivo, para cada um que for inválido. (0,1 ponto para cada identificador corretamente qualificado – se válido ou inválido e a justificativa quando for inválido)

IDENTIFICADOR	INVÁLIDO? (marque caso seja)	CASO SEJA INVÁLIDO, QUAL A EXPLICAÇÃO? (escreva a explicação nos espaços a seguir)
salário_bruto	<input checked="" type="checkbox"/>	É invalido devido o acento agudo.
_vezes	<input type="checkbox"/>	
valor INSS	<input checked="" type="checkbox"/>	É invalido devido o espaço em branco.
velocidade-maxima	<input checked="" type="checkbox"/>	É invalido devido ao caracter especial “ – “
Nota_1	<input type="checkbox"/>	

7. Por que, em algumas linguagens de programação como C, é necessário declarar as variáveis antes de seu uso? (0,5 ponto)

É necessário, para que o compilador possa verificar se as variáveis foram definidas corretamente antes de serem utilizadas, e para que o compilador reserve espaço na memória para a variável.

8. Na prática, qual a diferença entre declarar uma constante em C usando a diretiva de pré-processamento **#define** e usar uma variável declarada dentro da função **main()** como uma constante? Se você optar pela segunda forma, que cuidado deverá ter para que essa variável realmente se comporte como uma constante no programa? (0,5 ponto)

A diferença entre declarar uma constante usando a diretiva **#define** e declarar uma variável dentro da função **main()** como uma constante é que a primeira forma é resolvida pelo pré-processador e a segunda em tempo de compilação. Caso seja utilizada a segunda opção, é necessário ter cuidado para que a variável não seja modificada durante a execução do programa, para isso pode ser usada a palavra-chave “const”, que indica que a variável não pode ser modificada.

Nas questões a seguir será solicitado que você crie programas. Na avaliação das respostas será verificado:

- Se o programa funciona. Aproveite que você poderá usar um computador para criar os programas. Caso dê erro de compilação ou qualquer outro, será considerado errado e consequentemente sem pontuação na questão correspondente.
- Se o programa foi adequadamente comentado. Lembre-se da importância de usar comentários para documentar coisas importantes dentro do código fonte, inclusive com o nome do programa, sua finalidade, quem o criou e a data de criação. Programas não documentados o suficiente terão menos pontuação na questão correspondente.
- Se as variáveis utilizadas têm nomes significativos, de acordo com o contexto do programa, e não simplesmente letras como **a**, **b** ou **x**. Uma variável para o salário bruto pode ter o nome de **salario\_bruto** ou **sal\_bruto**, por exemplo, que são melhores do que **sb**. Programas com nomes de variáveis pouco significativas terão menos pontuação na questão correspondente.
- Se as variáveis possuem os tipos adequadamente definidos. Não vale criar todas as variáveis como **float**. Analise cada dado envolvido e defina o tipo da variável de acordo com os valores que serão usados. Programas que tenham variáveis com tipos mal definidos terão menos pontuação na questão correspondente.
- Se o código é legível e está bem-organizado, inclusive na indentação. Poucas coisas fazem sangrar os olhos como código mal organizado e de difícil leitura, uma verdadeira “porquice” (pobres porquinhos...). Programas com código mal organizado ou de difícil leitura terão menos pontuação na questão correspondente.
- Se as mensagens apresentadas ao usuário são explicativas. Por exemplo, se um programa vai calcular o novo valor de um salário a partir de um percentual fornecido, espera-se que as perguntas ao usuário sejam algo como *Qual o valor atual do salário?* e *Qual o*

*percentual de aumento?*, de forma que o usuário saiba exatamente o que deverá digitar. E, na hora de mostrar o novo valor do salário, deve-se usar uma mensagem como *Novo valor do salário: R\$* e o valor com duas casas decimais, ou então *O novo valor do salário é R\$*, seguido do novo valor com duas casas decimais. Se no começo do programa aparecer um texto explicando o que faz o programa, como por exemplo **CÁLCULO DO NOVO SALÁRIO**, ficará melhor ainda. Um bom programa tem uma boa interface com o usuário. Programas com mensagens pouco explicativas ou mal definidas terão menos pontuação na questão correspondente.

- Se os resultados obtidos ao executar o programa estão corretos. Não adianta não ter erros de compilação, o programa estar adequadamente comentado, você ter criado as variáveis com os tipos adequados e nomes significativos, que seu código esteja legível e bem-organizado e ter colocado mensagens explicativas ao usuário se ele apresenta resultados incorretos. Por exemplo, você criou um programa que tem um cálculo escrito incorretamente. Teste o programa com valores diferentes e verifique se os resultados apresentados são os que deveriam ser. Um bom programa é testado antes de ser entregue ao usuário. Programas que apresentem resultados incorretos terão menos pontuação na questão correspondente.

9. Crie um programa que peça ao usuário o valor atual do salário e o percentual de aumento do salário, que poderá ter casas decimais). Em seguida, o programa deverá exibir o valor do aumento (com duas casas decimais) e o novo valor do salário (com duas casas decimais). Por exemplo, se o salário atual for 1000 e o percentual de aumento for 10, o valor do aumento deverá ser 100 e o novo valor do salário deverá ser 1100. Salve esse programa como **prova9.cpp** e o entregue separado desse documento. Além disso, copie e cole o código fonte no espaço abaixo como indicado. (1,5 ponto)

```
#include <stdio.h>
#include <locale.h>

int main(){
    setlocale(LC_ALL, "");

    float sal_atual, sal_novo, porcentagem, acrescimo;

    printf("Informe seu salário atual:\n");
    scanf("%f", &sal_atual);
    printf("Insira a porcentagem de aumento:");
    scanf("%f", &porcentagem);

    acrescimo = sal_atual * (porcentagem / 100 );

    sal_novo = sal_atual + acrescimo;

    printf("O novo valor do salário é de R$%.2f", sal_novo);

    return 0;
}
```

10. Sabe-se que para calcular o perímetro de um triângulo é necessário somar as medidas de seus lados. Crie um programa que permita saber se o perímetro de um triângulo é pequeno (até 30 cm), médio (de 31 a 60 cm) ou grande (acima de 60 cm). Considere que as medidas dos lados são números inteiros. Salve esse programa como **prova10.cpp** e o entregue separado desse documento. Além disso, copie e cole o código fonte no espaço abaixo como indicado. (1,5 ponto)

```
#include <stdio.h>
```

```

#include <locale.h>

int main(){
    setlocale(LC_ALL, "");

    int lado_1, lado_2, lado_3, perimetro;

    printf("Informe o tamanho do lado 1:\n");
    scanf("%d", &lado_1);
    printf("Informe o tamanho do lado 2:\n");
    scanf("%d", &lado_2);
    printf("Informe o tamanho do lado 3:\n");
    scanf("%d", &lado_3);

    perimetro = lado_1 + lado_2 + lado_3;

    if(perimetro <= 30){
        printf("O triângulo é pequeno");
    }

    else if((perimetro >= 31) && (perimetro <=60)){
        printf("O triângulo é médio");
    }

    else if((perimetro > 60)){
        printf("O triângulo é grande");
    }

    printf("\n O perimetro é de %dcm", perimetro);

    return 0;
}

```

11. Crie um programa que exiba a mensagem **SALDO DEVEDOR** caso o saldo da conta corrente de um cliente seja negativo. Salve esse programa como **prova11.cpp** e o entregue separado desse documento. Além disso, copie e cole o código fonte no espaço abaixo como indicado. (1,5 ponto)

```

#include <stdio.h>
#include <locale.h>

int main(){
    setlocale(LC_ALL, "");

    float saldo;

    printf("Informe o saldo da sua conta corrente:\n");
    scanf("%f", &saldo);

    if(saldo < 0){
        printf("SALDO DEVEDOR");
    }

    else{
        printf("SALDO POSITIVO");
    }
}

```

```
}  
  
return 0;  
}
```

12. Foi definido que os funcionários de uma determinada empresa que tiverem salário bruto entre três e cinco salários mínimos serão selecionados para participar de uma pesquisa. Crie um programa que peça o valor do salário bruto (em reais) e exiba a mensagem **PARTICIPARÁ DA PESQUISA** caso deva ser selecionado para participar da pesquisa. Por outro lado, caso não deva ser selecionado para participar, deverá ser exibida a mensagem **NÃO PARTICIPARÁ DA PESQUISA**. Salve esse programa como **prova12.cpp** e o entregue separado desse documento. Além disso, copie e cole o código fonte no espaço abaixo como indicado. (1,5 ponto)

```
#include <stdio.h>  
#include <locale.h>  
  
int main(){  
    setlocale(LC_ALL, "");  
  
    float sal_bruto;  
  
    printf("Informe o seu salário bruto:\n");  
    scanf("%f", &sal_bruto);  
  
    if((sal_bruto > 3906) && (sal_bruto < 6510)){        // Salário mínimo R$1302,00  
        printf("PARTICIPARÁ DA PESQUISA");  
    }  
  
    else{  
        printf("NÃO PARTICIPARÁ DA PESQUISA");  
    }  
  
    return 0;  
}
```