

QUESTÃO 1

a)

```
questão 1.cpp
1 #include <iostream>
2
3 using namespace std;
4
5 // Função para calcular o fatorial
6 unsigned long long calcularFatorial(int numero) {
7     if (numero == 0 || numero == 1) {
8         return 1;
9     } else {
10         unsigned long long fatorial = 1;
11         for (int i = 2; i <= numero; ++i) {
12             fatorial *= i;
13         }
14         return fatorial;
15     }
16 }
17
18 int main() {
19     int num;
20
21     // Pedir ao usuário para inserir um número inteiro não negativo
22     cout << "Digite um numero inteiro nao negativo: ";
23     cin >> num;
24
25     // Verificar se o número é não negativo
26     if (num < 0) {
27         cout << "Numero negativo inserido. Por favor, insira um numero inteiro nao negativo." << endl;
28         return 1; // Sair do programa com erro
29     }
30
31     // Calcular e imprimir o fatorial
32     cout << "O fatorial de " << num << " eh: " << calcularFatorial(num) << endl;
33
34     return 0;
35 }
```

```
C:\Users\kamil\OneDrive\Doc x + v
Digite um numero inteiro nao negativo: 6
O fatorial de 6 eh: 720

-----
Process exited after 7.274 seconds with return value 0
Pressione qualquer tecla para continuar. . . |
```

b)

```
questão 1.cpp questão 2.cpp
1 #include <iostream>
2
3 using namespace std;
4
5 // Função para calcular o fatorial de um número
6 unsigned long long calcularFatorial(int numero) {
7     if (numero == 0 || numero == 1) {
8         return 1;
9     } else {
10         unsigned long long fatorial = 1;
11         for (int i = 2; i <= numero; ++i) {
12             fatorial *= i;
13         }
14         return fatorial;
15     }
16 }
17
18 int main() {
19     int numTermos = 10; // Número de termos para calcular
20     double e = 1.0; // Valor inicial de e
21
22     // Adicionar os termos à série
23     for (int i = 1; i <= numTermos; ++i) {
24         e += 1.0 / calcularFatorial(i);
25     }
26
27     // Imprimir o valor de e
28     cout << "O valor de e eh aproximadamente: " << e << endl;
29
30     return 0;
31 }
32
```

```
C:\Users\kamil\OneDrive\Doc x + v
O valor de e eh aproximadamente: 2.71828

-----
Process exited after 0.8162 seconds with return value 0
Pressione qualquer tecla para continuar. . . |
```

```

1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  // Função para calcular o fatorial de um número
7  unsigned long long calcularFatorial(int numero) {
8      if (numero == 0 || numero == 1) {
9          return 1;
10     } else {
11         unsigned long long fatorial = 1;
12         for (int i = 2; i <= numero; ++i) {
13             fatorial *= i;
14         }
15         return fatorial;
16     }
17 }
18
19 int main() {
20     int numTermos = 10; // Número de termos para calcular
21     double x; // Potência 'x'
22     double resultado = 1.0; // Valor inicial de e^x
23
24     // Pedir ao usuário para inserir a potência 'x'
25     cout << "Digite o valor da potencia 'x': ";
26     cin >> x;
27
28     // Adicionar os termos à série
29     for (int i = 1; i <= numTermos; ++i) {
30         resultado += pow(x, i) / calcularFatorial(i);
31     }
32
33     // Imprimir o valor de e^x
34     cout << "O valor de e^" << x << " é aproximadamente: " << resultado << endl;
35
36     return 0;
37 }

```

```
Process exited after 1.207 seconds with return value 0
Pressione qualquer tecla para continuar. . . |
```

[questão 1.cpp](#) [questão 2.cpp](#) [questão 3.cpp](#) [questão 4.cpp](#)

[illegible]

```
Process exited after 0.6942 seconds with return value 0
Pressione qualquer tecla para continuar. . . |
```

[questão 1.cpp](#)
[questão 2.cpp](#)
[questão 3.cpp](#)
[questão 4.cpp](#)
[questão 5.cpp](#)

```

4 using namespace std;
5
6 int main() {
7     cout << "Triplos de Pitágoras para lados menores ou iguais a 20:" << endl;
8
9     // Laço triplamente aninhado para tentar todas as possibilidades
10    for (int lado1 = 1; lado1 <= 20; ++lado1) {
11        for (int lado2 = 1; lado2 <= 20; ++lado2) {
12            for (int hipotenusa = 1; hipotenusa <= 20; ++hipotenusa) {
13                // Verificar se a relação de Pitágoras é satisfeita
14                if (lado1 * lado1 + lado2 * lado2 == hipotenusa * hipotenusa) {
15                    // Imprimir o triplo de Pitágoras encontrado
16                    cout << "Lados: " << lado1 << ", " << lado2 << ", Hipotenusa: " << hipotenusa << endl;
17                }
18            }
19        }
20    }
21
22    return 0;
23 }

```

```
Process exited after 2.534 seconds with return value 0
Pressione qualquer tecla para continuar. . . |
```

```
questão 1.cpp questão 2.cpp questão 3.cpp questão 4.cpp questão 5.cpp questão 6.cpp
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     const double precoProduto1 = 2.98;
6     const double precoProduto2 = 4.58;
7     const double precoProduto3 = 9.98;
8     const double precoProduto4 = 4.49;
9     const double precoProduto5 = 6.87;
10    int numeroProduto;
11    int quantidadeVendida;
12    double totalVendas = 0.0;
13    while (true) {
14        cout << "Digite o número do produto (1-5) ou -1 para parar: ";
15        cin >> numeroProduto;
16        if (numeroProduto == -1) {
17            break;
18        }
19        cout << "Digite a quantidade vendida: ";
20        cin >> quantidadeVendida;
21
22        switch (numeroProduto) {
23            case 1:
24                totalVendas += precoProduto1 * quantidadeVendida;
25                break;
26            case 2:
27                totalVendas += precoProduto2 * quantidadeVendida;
28                break;
29            case 3:
30                totalVendas += precoProduto3 * quantidadeVendida;
31                break;
32            case 4:
33                totalVendas += precoProduto4 * quantidadeVendida;
34                break;
35            case 5:
36                totalVendas += precoProduto5 * quantidadeVendida;
37                break;
38            default:
39                cout << "Número de produto inválido. Por favor, tente novamente." << endl;
40                break;
41        }
42    }
43    cout << "O total de vendas eh: R$ " << totalVendas << endl;
44}
```

```
C:\Users\kamili\OneDrive\Doc x + v
Digite o número do produto (1-5) ou -1 para parar: 4
Digite a quantidade vendida: -1
Digite o número do produto (1-5) ou -1 para parar: -1
0 total de vendas eh: R$ -4.49

-----
Process exited after 9.747 seconds with return value 0
Pressione qualquer tecla para continuar. . . |
```

Questão 5

```
questão 1.cpp questão 2.cpp questão 3.cpp questão 4.cpp questão 5.cpp questão 6.cpp questão 7.cpp
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     const int tamanhoMaximo = 30; // Tamanho máximo para os números
7     const int quantidadeNumeros = 5; // Quantidade de números a serem lidos
8
9     int numeros[quantidadeNumeros];
10
11    // Solicitar ao usuário para inserir os números
12    cout << "Digite cinco numeros entre 1 e 30:" << endl;
13    for (int i = 0; i < quantidadeNumeros; ++i) {
14        cout << "Numero " << i + 1 << ": ";
15        cin >> numeros[i];
16    }
17
18    // Verificar se o número está dentro do intervalo permitido
19    if (numeros[i] < 1 || numeros[i] > tamanhoMaximo) {
20        cout << "Numero inválido. Por favor, insira um numero entre 1 e 30." << endl;
21        --i; // Volta para a iteração anterior para tentar novamente
22    }
23
24    // Exibir as barras de asteriscos
25    cout << "nGrafico de barras:" << endl;
26    for (int i = 0; i < quantidadeNumeros; ++i) {
27        cout << "Numero " << i + 1 << ": ";
28        for (int j = 0; j < numeros[i]; ++j) {
29            cout << " ";
30        }
31        cout << endl;
32    }
```

```
C:\Users\kamili\OneDrive\Doc x + v
Digite cinco numeros entre 1 e 30:
Numero 1: 6
Numero 2: 7
Numero 3: 5
Numero 4: 5
Numero 5: 7

Grafico de barras:
Numero 1: *****
Numero 2: *****
Numero 3: *****
Numero 4: *****
Numero 5: *****

-----
Process exited after 10.16 seconds with return value 0
Pressione qualquer tecla para continuar. . . |
```

Questão 6

questão 1.cpp questão 2.cpp questão 3.cpp questão 4.cpp questão 5.cpp questão 6.cpp questão 7.cpp questão 8.cpp

```

6
7 int main() {
8     srand(time(0)); // Semente para geração de números aleatórios
9
10    int limiteInferior = 1; // Limite inferior do intervalo de números
11    int limiteSuperior = 100; // Limite superior do intervalo de números
12    int numero; // Número a ser adivinhado pelo programa
13
14    // Gerar um número aleatório para ser adivinhado pelo programa
15    numero = rand() % 100 + 1; // Número entre 1 e 100
16
17    cout << "Pense em um numero entre 1 e 100 e vou tentar adivinha-lo em no maximo sete perguntas." << endl;
18
19    // Loop para fazer no máximo sete perguntas
20    for (int tentativa = 1; tentativa <= 7; ++tentativa) {
21        int palpite = (limiteInferior + limiteSuperior) / 2; // Palpite do programa (média do intervalo)
22
23        cout << "O numero em que voce esta pensando e menor que " << palpite << "? (Digite 's' para sim, 'n' para nao): ";
24        char resposta;
25        cin >> resposta;
26
27        if (resposta == 's') {
28            limiteSuperior = palpite - 1; // Ajusta o limite superior para o palpite - 1
29        } else {
30            limiteInferior = palpite + 1; // Ajusta o limite inferior para o palpite + 1
31        }
32
33        // Verificar se o palpite do programa é igual ao número pensado pelo usuário
34        if (palpite == numero) {
35            cout << "O numero que voce pensou e " << palpite << ". Eu acertei!" << endl;
36            break; // Encerra o jogo
37        }
38    }
39
40    if (limiteInferior > limiteSuperior) {
41        cout << "Voce trapaceou! O numero que voce pensou nao esta no intervalo correto." << endl;
42    } else if (limiteInferior <= limiteSuperior) {
43        cout << "Nao consegui adivinhar o numero em sete tentativas. Voce ganhou!" << endl;
44    }
45
46    return 0;
47 }

```

```

C:\Users\kamill\OneDri x + - □ ×
Pense em um numero entre 1 e 100 e vou tenta
r adivinha-lo em no maximo sete perguntas.
O numero em que voce esta pensando e menor q
ue 50? (Digite 's' para sim, 'n' para nao):
s
O numero em que voce esta pensando e menor q
ue 25? (Digite 's' para sim, 'n' para nao):
s
O numero em que voce esta pensando e menor q
ue 12? (Digite 's' para sim, 'n' para nao):
s
O numero em que voce esta pensando e menor q
ue 6? (Digite 's' para sim, 'n' para nao): s
O numero em que voce esta pensando e menor q
ue 3? (Digite 's' para sim, 'n' para nao): s
O numero em que voce esta pensando e menor q
ue 1? (Digite 's' para sim, 'n' para nao): n
O numero em que voce esta pensando e menor q
ue 2? (Digite 's' para sim, 'n' para nao): n
Voce trapaceou! O numero que voce pensou nao
esta no intervalo correto.

-----
Process exited after 12.19 seconds with retu
rn value 0
Pressione qualquer tecla para continuar. . .
|

```