

[Painel](#) / [Meus cursos](#) / [GRADUAÇÃO](#) / [CIÊNC. DA COMPUTAÇÃO](#) / [CCP130](#) / [Projeto e Provas](#)
/ [Avaliação PV - 26/11/2021 - 15:00](#)

Iniciado em sexta, 26 Nov 2021, 15:00

Estado Finalizada

Concluída em sexta, 26 Nov 2021, 16:26

**Tempo
empregado** 1 hora 26 minutos

Avaliar **7,00** de um máximo de 10,00(70%)



Questão 1

Não respondido

Atingiu 0,00 de 3,00

Um microcontrolador realiza a medição de tempo gasto em um conjunto de determinadas tarefas salvando os valores em arquivos txt. O tempo salvo pelo microcontrolador está em **segundos**. O primeiro valor que o microcontrolador salva é o tempo em que ele está ligado; todos os valores registrados na sequência são os tempos nos momentos em que as tarefas são concluídas. Assim, o tempo gasto por cada tarefa pode ser encontrado pela diferença de tempo ($T_f - T_i$), onde T_i é o valor da linha anterior e T_f é o valor de cada registro lido.

Escreva uma função chamada **leituraTempo()** que recebe como parâmetro o nome de um arquivo e realiza o cálculo do tempo gasto por cada tarefa em **minutos**, imprimindo os valores com o tipo inteiro. Os arquivos já estão previamente adicionados no Moodle para os testes; um exemplo de arquivo pode ser encontrado aqui: "[tempo.txt](#)" (clique com o botão direito em cima do link e em "Salvar link como..." para baixar o arquivo).

Atenção: Desenvolver somente a função, caso contrário a questão será zerada pelo sistema.

For example:

Test	Result
#include <stdio.h>	6
	3
int main()	6
{	21
leituraTempo("tempo.txt");	25
return 0;	
}	

Answer: (penalty regime: 0 %)

1 ||

Question author's solution (C):

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void leituraTempo(char nome[]){
5      int tempo_inicial = 0, tempo_lido = 0, tempo=0;
6      FILE *fr = fopen(nome,"r");
7      fscanf(fr, "%d", &tempo_inicial);
8
9      while (fscanf(fr, "%d", &tempo_lido) != EOF)
10     {
11         tempo = (tempo_lido - tempo_inicial)/60;
12         printf("%d\n", tempo);
13         tempo_inicial = tempo_lido;

```



Questão **2**

Correto

Atingiu 0,50 de 0,50

Considere o código abaixo:

```
#include <stdio.h>
#include <math.h>
int f(int *v1, int *v2){
    int x = 0;
    for(int i = 0; i < 4; i++){
        x += v1[i]*v2[i];
    }
    return x;
}
int main(){
    int x, v1[4] = {1, -1, 2, -2}, v2[4] = {1, 2, 3, 4};
    x = f(v1, v2);
    printf("x = %d ", x);
}
```

A saída desse código será:

- ☐ a. x = -1
- ☐ b. x = 0
- ☐ c. NDA
- ☒ d. x = -3
- ☐ e. x = 6



Sua resposta está correta.

A resposta correta é: x = -3



Questão 3

Correto

Atingiu 1,00 de 1,00

Leia um valor **X**. Coloque este valor na primeira posição de um vetor **N[10]**. Em cada posição subsequente de N (1 até 9), coloque o dobro do valor armazenado na posição anterior, conforme o exemplo abaixo. Imprima o vetor **N**.

For example:

Input	Result
5	N[0] = 5 N[1] = 10 N[2] = 20 N[3] = 40 N[4] = 80 N[5] = 160 N[6] = 320 N[7] = 640 N[8] = 1280 N[9] = 2560

Answer: (penalty regime: 0 %)

```

1  /*
2  AUTOR: FRITZ
3  DATA: 26/11/2021
4
5  Leia um valor X. Coloque este valor na primeira posição de um vetor N[10].
6      Em cada posição subsequente de N (1 até 9),
7          coloque o dobro do valor armazenado na posição anterior, conforme o exemplo abaixo.
8      Imprima o vetor N.
9  */
10
11 #include <stdio.h>
12 #include <stdlib.h>
13

```

	Input	Expected	Got	
✓	5	N[0] = 5 N[1] = 10 N[2] = 20 N[3] = 40 N[4] = 80 N[5] = 160 N[6] = 320 N[7] = 640 N[8] = 1280 N[9] = 2560	N[0] = 5 N[1] = 10 N[2] = 20 N[3] = 40 N[4] = 80 N[5] = 160 N[6] = 320 N[7] = 640 N[8] = 1280 N[9] = 2560	✓
✓	3	N[0] = 3 N[1] = 6 N[2] = 12 N[3] = 24 N[4] = 48 N[5] = 96 N[6] = 192 N[7] = 384 N[8] = 768 N[9] = 1536	N[0] = 3 N[1] = 6 N[2] = 12 N[3] = 24 N[4] = 48 N[5] = 96 N[6] = 192 N[7] = 384 N[8] = 768 N[9] = 1536	✓



	Input	Expected	Got	
✓	10	N[0] = 10 N[1] = 20 N[2] = 40 N[3] = 80 N[4] = 160 N[5] = 320 N[6] = 640 N[7] = 1280 N[8] = 2560 N[9] = 5120	N[0] = 10 N[1] = 20 N[2] = 40 N[3] = 80 N[4] = 160 N[5] = 320 N[6] = 640 N[7] = 1280 N[8] = 2560 N[9] = 5120	✓
✓	7	N[0] = 7 N[1] = 14 N[2] = 28 N[3] = 56 N[4] = 112 N[5] = 224 N[6] = 448 N[7] = 896 N[8] = 1792 N[9] = 3584	N[0] = 7 N[1] = 14 N[2] = 28 N[3] = 56 N[4] = 112 N[5] = 224 N[6] = 448 N[7] = 896 N[8] = 1792 N[9] = 3584	✓

Passou em todos os teste! ✓

Question author's solution (C):

```

1  #include <stdio.h>
2
3  int main ()
4  {
5      int i;
6      int x[10];
7      int n;
8
9      scanf("%d", &n);
10     x[0] = n;
11     for (i = 1; i < 10; i++)
12     {
13         n = n*2;

```

Correto

Notas para este envio: 1,00/1,00.



Questão 4

Correto

Atingiu 2,00 de 2,00

Escreva um programa que calcule a e^x , sabendo que:

$$e^x \approx 1 + \frac{x}{1!} + \frac{(x)^2}{2!} + \frac{(x)^3}{3!} + \frac{(x)^4}{4!} + \frac{(x)^5}{5!} + \dots$$

Considere somente os 7 primeiros termos da equação na sua solução.

For example:

Input	Result
3.5	Digite x: exp(3.50) = 30.953407

Answer: (penalty regime: 0 %)

Reset answer

```

1 #include <stdio.h>
2 #include <math.h>
3
4 int fatorial(int n){
5     // FAZER
6     int fatorial=1;
7     while(n>0){
8         fatorial *= n;
9         n --;
10    }
11    return fatorial;
12 }
13
14 float exponencial(float base, int expoente){
15     // FAZER
16     float exxp;
17     exxp = pow(base, expoente);
18     return exxp;
19 }
20
21 int main(){
22     // FAZER
23     // chame a função exponencial
24     // exiba o resultado
25     float x;
26     float resultado = 0;

```

	Input	Expected	Got	
✓	3.5	Digite x: exp(3.50) = 30.953407	Digite x: exp(3.50) = 30.953407	✓
✓	1.7	Digite x: exp(1.70) = 5.463684	Digite x: exp(1.70) = 5.463684	✓
✓	2	Digite x: exp(2.00) = 7.355556	Digite x: exp(2.00) = 7.355556	✓
✓	0.5	Digite x: exp(0.50) = 1.648720	Digite x: exp(0.50) = 1.648720	✓

Passou em todos os testes! ✓

Question author's solution (C):

```

1 #include <stdio.h>
2 #include <math.h>
3
4 int fatorial(int n){
5     if(n <= 0)
6         return 1;

```



```
7 |         return(n*fatorial(n-1));
8 |     }
9 |
10 | float exponencial(float xx){
11 |     float exp = 0.0;
12 |     for(int i = 0 ; i < 7 ; i++)
13 |         exp += pow(xx,i)/fatorial(i) ;
14 |     return exp;
15 | }
16 |
17 |
18 | int main(){
19 |     float x ;
20 |     printf("Digite x: ");
21 |     scanf("%f", &x);
22 |     printf("\nexp(%.2f) = %.6f",x,exponencial(x));
23 |     return 0;
24 | }
25 |
```

Correto

Notas para este envio: 2,00/2,00.

Questão 5

Correto

Atingiu 0,50 de 0,50

Qual será a saída do seguinte código em C?

```
#include <stdio.h>
int main()
{
    int *ptr, a = 10;
    ptr = &a;
    *ptr += 0;
    printf("%d,%d\n", *ptr, a);
}
```

Escolha uma opção:

- ☐ a. Endereço de a,10
- ☐ b. 11,10
- ☐ c. 10,11
- ☐ d. 11,11
- ☒ e. 10,10



Sua resposta está correta.

A resposta correta é: 10,10



Questão 6

Correto

Atingiu 0,50 de 0,50

A função main abaixo está errada! A ideia é ter a variável **x** e o ponteiro **p**, que deve apontar para **x**.

```
1 void main() {  
2     int x, *p;  
3     x = 100;  
4     p = x;  
5     printf("Valor de p: %d\n", *p);  
6 }
```

Na linha 5, deve ser exibido o valor armazenado no endereço para o qual **p** aponta (valor 100). Considerando que você precisa corrigir o código, quais seriam as linhas com necessidade de alteração?

Escolha uma opção:

- ☐ a. 2 somente
- ☐ b. 2 e 4
- ☒ c. 4 somente
- ☐ d. 3 e 5
- ☐ e. 2, 4 e 5



Sua resposta está correta.

A resposta correta é: 4 somente

Questão 7

Correto

Atingiu 0,50 de 0,50

Supondo que os elementos do vetor **v** são do tipo **int** e cada **int** ocupa 2 bytes no seu computador.

Se o endereço de **v[0]** é 10000, qual o valor da expressão **v + 3** (qual será o endereço. Dez mil e ...)?

Escreva sua resposta em números inteiros sem nenhuma pontuação.

Resposta:



A resposta correta é: 10006



Questão 8

Correto

Atingiu 2,00 de 2,00

Crie uma *struct* para representar um Ponto no espaço 2D (int x, y). Crie também a função *coincidentes(...)* que tem como parâmetro um vetor de Pontos (que vai receber 2 pontos). A função deve retornar 1 caso os dois pontos sejam coincidentes e 0 caso contrário. A função *main* já é dada pelo Moodle no caso de teste. Você deve criar a *struct* e a função *coincidentes(...)* somente.

For example:

Test	Result
<pre>struct Ponto pontos[2]; pontos[0].x = 2; pontos[0].y = 3; pontos[1].x = 4; pontos[1].y = 5; if(coincidentes(pontos)) puts("Os pontos são coincidentes"); else puts("Os pontos não são coincidentes");</pre>	Os pontos não são coincidentes
<pre>struct Ponto pontos[2]; pontos[0].x = 2; pontos[0].y = 5; pontos[1].x = 2; pontos[1].y = 5; if(coincidentes(pontos)) puts("Os pontos são coincidentes"); else puts("Os pontos não são coincidentes");</pre>	Os pontos são coincidentes

Answer: (penalty regime: 0 %)

```
1  /*
2  AUTOR: FRITZ
3  DATA: 26/11/2021
4
5  Crie uma struct para representar um Ponto no espaço 2D (int x, y).
6      Crie também a função coincidentes(...) que tem como parâmetro
7          um vetor de Pontos (que vai receber 2 pontos).
8
9  A função deve retornar 1 caso os dois pontos sejam coincidentes e 0 caso contrário.
10     A função main já é dada pelo Moodle no caso de teste.
11     Você deve criar a struct e a função coincidentes(...) somente.
12  */
13 struct Ponto{
```

	Test	Expected	Got	
✓	<pre>struct Ponto pontos[2]; pontos[0].x = 2; pontos[0].y = 3; pontos[1].x = 4; pontos[1].y = 5; if(coincidentes(pontos)) puts("Os pontos são coincidentes"); else puts("Os pontos não são coincidentes");</pre>	Os pontos não são coincidentes	Os pontos não são coincidentes	✓



	Test	Expected	Got	
✓	<pre> struct Ponto pontos[2]; pontos[0].x = 2; pontos[0].y = 5; pontos[1].x = 2; pontos[1].y = 5; if(coincidentes(pontos)) puts("Os pontos são coincidentes"); else puts("Os pontos não são coincidentes"); </pre>	Os pontos são coincidentes	Os pontos são coincidentes	✓
✓	<pre> struct Ponto pontos[2]; pontos[0].x = 10; pontos[0].y = 4; pontos[1].x = 9; pontos[1].y = 1; if(coincidentes(pontos)) puts("Os pontos são coincidentes"); else puts("Os pontos não são coincidentes"); </pre>	Os pontos não são coincidentes	Os pontos não são coincidentes	✓
✓	<pre> struct Ponto pontos[2]; pontos[0].x = 3; pontos[0].y = 7; pontos[1].x = 3; pontos[1].y = 7; if(coincidentes(pontos)) puts("Os pontos são coincidentes"); else puts("Os pontos não são coincidentes"); </pre>	Os pontos são coincidentes	Os pontos são coincidentes	✓

Passou em todos os teste! ✓

Question author's solution (C):

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4
5  struct Ponto {
6      int x;
7      int y;
8  };
9
10 int coincidentes(struct Ponto pontos[]) {
11     if(pontos[0].x == pontos[1].x && pontos[0].y == pontos[1].y)
12         return 1;
13     else

```

Correto

Notas para este envio: 2,00/2,00.

◀ URI

Seguir para...

Projeto IAC ▶

