



Nome: Pedro Henrique Camabarú Data: 28/10/25

A solução dos problemas propostos deverá ser apresentada por meio de programas escritos na linguagem C.
Para avaliação das soluções serão considerados os critérios: lógica da solução, identação do código e uso da linguagem de programação C.

Leia atentamente cada questão. Boa avaliação!!!

1. (0,9) Uma empresa de taxi necessita manter informatizado o cadastro de seus 50 veículos. Para cada um, são armazenados os seguintes dados: placa, ano, modelo, valor do IPVA em 2025.

- ~~a. Defina a estrutura de dados~~ necessária para armazenar os dados de cada veículo neste Sistema.
~~b. Defina a variável~~ necessária para armazenar todos os dados dos 50 veículos.
~~c. Especifique a sequência de código~~ que deverá ser feita para calcular o valor total do IPVA em 2025, considerando as definições feitas nas questões 1.a e 1.b.

2. (0,4) Considere que o endereço de uma variável Y foi atribuído a um ponteiro pY. Quais das seguintes expressões são verdadeiras?

- ~~a. pY == &Y~~
~~b. Y == &pY~~ → desconsiderar
~~c. Y == *pY~~
d. pY == *Y

resposta final: a) e c)

3. (0,4) Considere a variável inteira x e o ponteiro para inteiro px que recebeu o endereço de x. Quais instruções são corretas para fazer a leitura do valor de x?

- ~~a. scanf("%d", *px);~~
~~b. scanf("%d", px);~~
~~c. scanf("%d", &x);~~
d. scanf("%d", x); → desconsiderar, resposta final: b) e c)

4. (0,4) Considere o trecho de código:

```
int x, *px;  
float a, *pa;
```

Ques

5. (0,4) Considere o trecho de código:

```
int *p;  
int a = 5;  
p = &a;
```

Assinale V (verdadeiro) ou F (falso), conforme a veracidade de cada sentença:

- ~~(F) *p é igual a 5.~~
~~(V) p é igual a 5.~~
~~(V) p armazena o endereço de a.~~
~~(F) ao executar *p = 30, a terá o valor 30.~~
~~(F) ao ser alterado o valor de a, *p será modificado.~~
~~(V) ao ser alterado o valor de a, o p será modificado.~~

Ques 5
Quais atribuições são permitidas? Assinale V (verdadeiro) ou F (falso)

- ~~(V) x = 10; V~~
~~(F) *pa = &a; F~~
~~(F) px = &a; F~~
~~(V) *px = 2; V~~
~~(V) px = &x; V~~
~~(F) pa = &x; F~~
~~(F) *px = 3.5; F~~
~~(V) *pa = 4.5; V~~

6. (0,4) Considere o código a seguir. Mostre o que será exibido como saída na tela, quando este código for executado.

Q25

```

1 #include <stdio.h>
2
3 void funcao1(int *v){
4     int i;
5     for(i = 0; i<5; i++){
6         *(v+i) = i;
7     }
8     return ;
9 }
10
11 int funcao2(int *v, int *r){
12     int i;
13     for(i = 0; i<5; i++){
14         *(r+i) = *(v+i) * *(v+i);
15     }
16     return i;
17 }
18
19 void funcao3(int *v){
20     int i;
21     for(i = 0; i<5; i++){
22         printf("%d ", *(v+i));
23     }
24     return ;
25 }
26
27 int main(){
28     int a[5], b, c[5];
29
30     funcao1(a);
31     b = funcao2(a, c);
32     printf("b = %d\n", b);
33     funcao3(c);
34
35     return 0;
36 }
37
38

```

$$a = [0, 1, 2, 3, 4]$$

$$c = [0, 1, 2, 3, 4]$$

b = [0, 1, 4, 9, 16]

~~e~~

~~b=5~~

~~é um~~

7. (0,4) Considere o algoritmo recursivo a seguir. Mostre o que será exibido como saída na tela, quando este código for executado.

Q25

```

1 #include <stdio.h>
2
3 void calcula(int x){
4     if (x == 1)
5         printf("%d ", x);
6     else{
7         printf("%d ", x);
8         calcula(x-1);
9     }
10    return ;
11 }
12
13
14 int main(){
15     int a;
16
17     printf("Insira o número do mês de seu aniversário: ");
18     scanf("%d", &a);
19
20     if (a >=1 && a<=12)
21         calcula(a);
22     else
23         printf("Insira um valor do intervalo [1, 12]");
24
25     return 0;
26 }

```

mês aniversário: 11

$$\text{calcula}(11 - 1) = 10$$

11 10 9 8... 1

O resultado será 10.

8. (0,4) Considere o exemplo abaixo, o que será produzido como saída em tela, pelas linhas 25 e 26, se for digitado, como palavra de entrada, o seu primeiro nome?

Q35

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int mostra(char *p){
5     int a=0;
6
7     for( ; *p; p++){
8         a++;
9         if (*p == 'a' || *p == 'e' || *p == 'i')
10             *p = '#';
11         else if (*p == 'o' || *p == 'u')
12             *p = 'x';
13     }
14
15     return a;
16 }
17 main() {
18     char s[30];
19     int b;
20
21     scanf("%s", s);
22
23     b = mostra(s);
24
25     printf("b = %d\n", b);
26     printf("%s", s);
27
28 }
29 }
```

nome : pedro

resultado : p#dx

e

b=5

9. (0,6) Na resolução desta questão, utilize **funções e ponteiros**:

Q3 QK

Escreva um algoritmo para ler uma *string*. Desenvolva uma função que mostra a posição de cada vogal encontrada na *string*. A função deve retornar a quantidade de letras vogais na *string*. A função *main* deve exibir essa quantidade. Utilize ponteiros para percorrer a *string*.

10. (1,2) Na resolução desta questão, utilize **funções e ponteiros**. Um vetor de inteiros, com 31 posições, armazena a quantidade de acessos ao Portal do Vestibular, no mês de Outubro/2025. Desenvolva:

Q2

a. uma função para ler os dados desse vetor.

b. uma função para calcular e retornar a média de acessos ao portal no mês de outubro/2025.

Posteriormente, leia a média de acessos ao Portal do Vestibular do mês de outubro/2024 e desenvolva:

c. uma função para mostrar em qual dos dois ~~meses~~ houve maior quantidade de acessos e qual foi essa diferença, em percentual.

anos

Q1

11. (0,5) Em uma faculdade, o *login* dos usuários é gerado automaticamente pelo sistema e corresponde às letras iniciais do nome de cada aluno. Faça um algoritmo para ler uma *string* que contém o nome de um aluno e produza uma nova *string* com o login. Utilize ponteiros para referenciar a *string*.