



PARA CADA UMA DAS QUESTÕES QUE REQUEREM A ESCRITA DE UMA OU MAIS FUNÇÕES, ESCREVA TAMBÉM UM PROGRAMA PRINCIPAL (A FUNÇÃO MAIN) QUE CHAME A(S) FUNÇÃO(ÕES).

PARTE 1

- 1 Crie um programa que declare duas variáveis (inteiro e real), atribua valores para elas e crie dois ponteiros, cada um apontando para uma destas variáveis. Mostre na tela: O valor e o endereço de memória das variáveis, o valor do ponteiro, o endereço do ponteiro e o valor apontado pelo ponteiro. Quais destes valores são iguais? Justifique.
- 2 Escreva uma função **troca** com o seguinte protótipo: **void troca (int*, int*)**. Escreva um programa que receba dois valores inteiros do teclado e chame a função **troca**. O programa deve imprimir os valores antes e depois da troca.
- 3 Utilizando a função **troca** desenvolvida na questão anterior, escreva uma função que troque o valor de duas variáveis do tipo inteiro **SOMENTE** se o valor da primeira variável for **MENOR** que o valor da segunda variável. Ou seja, o maior valor passará a ser a primeira variável.

A função deve possuir o seguinte protótipo:

```
void trocaSeMenor(int* x, int* y);
```

- 4 Marcos deve organizar três pilhas de caixas em ordem decrescente de altura. Faça um programa receba como entrada três valores correspondendo à altura das pilhas e imprima na tela as alturas em ordem decrescente.

Importante: Este programa deve necessariamente utilizar somente três variáveis do tipo inteiro (uma para cada altura de pilha) e utilizar a função **trocaSeMenor** desenvolvida na questão anterior.

Exemplos de Entradas e Saídas.

Entrada	Saída Esperada
54 99 23	99 54 23
100 50 3	100 50 3
9 20 55	55 20 9
60 7 70	70 60 7

- 5 Explique o que as linhas abaixo fazem.

(a)

```
int i=99, j;  
int *p;  
p = &i;  
j = *p + 100;
```

(b)

```
int a=5, b=12;  
int *p;  
int *q;  
p = &a;  
q = &b;  
int c = *p + *q;
```

(c)

```
int i=7, j=3;  
int *p;  
int **r;  
p = &i;  
r = &p;  
c = **r + j;
```

- 6 Quais serão os valores de x, y e p ao final do trecho de código abaixo?

```
int x, y, *p;  
y = 0;  
p = &y;  
x = *p;  
  
x = 4;  
(*p)++;  
--x;  
(*p) += x;
```

7 O programa (trecho de código) abaixo possui erros. Qual(is)? Como deveriam ser?

```
(a) void main() {  
    int x, *p;  
    x = 100;  
    p = x;  
    cout << "Valor de  
    p:\n" << *p;  
}
```

8 Qual o resultado do código abaixo? Explique cada linha.

```
int x = 100, *p, **pp;  
p = &x;  
pp = &p;  
cout<<"Valor de pp: " << **pp;
```

9 O que significa o operador asterisco em cada um dos seguintes casos:

- a) `int *p;`
- b) `cout << *p;`
- c) `*p = x*5;`
- d) `cout << *(p+1);`

10 Qual é a saída deste programa?

```
#include void main( ) {  
    int i=5, *p;  
    p = &i;  
    cout<< p << '\t' << (*p+2) << '\t' << **&p << '\t' << 3**p << '\t' << (**&p+4);  
}
```

11 Seja o seguinte trecho de programa:

```
int i=3, j=5;  
int *p, *q;  
p = &i;  
q = &j;
```

Qual é o valor das seguintes expressões?

- a) `p == &i;`
- b) `*p - *q`
- c) `**&p`
- d) `3* - *p / (*q) + 7`

12 Qual será a saída deste programa supondo que **i** ocupa o endereço **4094** na memória?

```
int main() {  
    int i=5, *p;  
    p = &i;  
    cout << p << *p+2 << **&p << 3**p << **&p+4;  
}
```

13 Se **i** e **j** são variáveis inteiras, e **p** e **q** ponteiros para `int`, quais das seguintes expressões de atribuição são ilegais?

- a) `p = &i;`
- b) `*q = &j;`
- c) `p = *&i;`
- d) `i = (*&)j;`
- e) `i = *&j;`
- f) `i = *&*&j;`
- g) `q = *p;`
- h) `i = (*p)++ + *q`

14 Qual serão as saídas do seguinte programa? Após responder, execute-o e verifique se foram obtidas as respostas esperadas.

```
#include <iostream>  
using namespace std;  
int main() {  
    int valor;  
    int *p1;  
    float temp;  
    float *p2;  
    char aux;
```

```

char *nome = "Algoritmos";
char *p3;
int idade;
int vetor[3];
int *p4;
int *p5;
/* (a) */
valor = 10;
p1 = &valor;
*p1 = 20;
cout << " (a) " << valor << endl;
/* (b) */
temp = 26.5;
p2 = &temp;
*p2 = 29.0;
cout << " (b) " << temp << endl;
/* (c) */
p3 = &nome[0];
aux = *p3;
cout << " (c) " << aux << endl;
/* (d) */
p3 = &nome[4];
aux = *p3;
cout << " (d) " << aux << endl;
/* (e) */
p3 = nome;
cout << " (e) " << *p3 << endl;
/* (f) */
p3 = p3 + 4;
cout << " (f) " << *p3 << endl;
/* (g) */
p3--;
cout << " (g) " << *p3 << endl;
/* (h) */
vetor[0] = 31;
vetor[1] = 45;
vetor[2] = 27;
p4 = vetor;
idade = *p4;
cout << " (h) " << idade << endl;
/* (i) */
p5 = p4 + 1;
idade = *p5;
cout << " (i) " << idade << endl;
/* (j) */
p4 = p5 + 1;
idade = *p4;
cout << " (j) " << idade << endl;
/* (l) */
p4 = p4 - 2;
idade = *p4;
cout << " (l) " << idade << endl;
/* (m) */
p5 = &vetor[2] - 1;
cout << " (m) " << *p5 << endl;
/* (n) */
p5++;
cout << " (n) " << *p5 << endl;
return(0);
}

```

- 15 Crie um programa para calcular a área e o perímetro de um hexágono. O seu programa deve implementar uma função chamada `calcula_hexagono` que calcule a área e o perímetro de um hexágono regular de lado l com o seguinte protótipo:

```
void calcula_hexagono(float l, float *area, float *perimetro);
```

Lembre que a área e o perímetro de um hexágono regular são dados por:

$$A = \frac{3l^2\sqrt{3}}{2} \quad P = 6l$$

Em seguida crie a função principal do programa e utilize a função `calcula_hexagono` para calcular a área e o perímetro de um hexágono de lado l informado pelo usuário.

- 16 Escreva uma função que determine a média e a situação de um aluno em uma disciplina. A função recebe como parâmetros as três notas de um aluno (p_1 , p_2 , e p_3), seu número de faltas ($faltas$), o número total de aulas da disciplina ($aulas$) e o ponteiro para uma variável ($media$), conforme o seguinte protótipo:

```
char situacao(float p1, float p2, float p3, int faltas, int aulas, float *media);
```

Na variável indicada pelo ponteiro $media$, a função deve armazenar a média do aluno, calculada como a média aritmética das três provas. Além disso, a função deve retornar um caractere indicando a situação do aluno no curso, definido de acordo com o seguinte critério:

Número de Faltas	Média	Situação	Retorno
$\leq 25\%$ do total de aulas	$\geq 7,0$	Aprovado	A
	$< 7,0$	Reprovado	R
$> 25\%$ do total de aulas	Qualquer	Reprovado por faltas	F

Em seguida, escreva a função principal de um programa que utiliza a função anterior para determinar a situação de um aluno. O programa deve:

- Ler do teclado três números reais e dois números inteiros, representando as notas da p_1 , p_2 e p_3 , o número de faltas e o número de aulas, respectivamente;
 - Chamar a função desenvolvida na primeira questão para determinar a média e a situação do aluno na disciplina;
 - Exibir a média (com apenas uma casa decimal) e a situação do aluno, isto é, "APROVADO", "REPROVADO" ou "REPROVADO POR FALTAS", dependendo do caractere retornado pela função, conforme a tabela acima.
- 17 Determine o que será mostrado pelo seguinte programa. Execute-o e verifique se foram obtidas as respostas esperadas.
- ```
1. #include <iostream>
2. using namespace std;
3.
4. int main(){
5. float vet[5] = {1.1, 2.2, 3.3, 4.4, 5.5};
6. float *f;
7. int i;
8. f = vet;
9. cout << "contador/valor/valor/endereco/endereco" << endl << endl;
10. for(i = 0 ; i <= 4 ; i++){
11. cout << "\ni = " << i << endl;
12. cout << "vet[" << i << "] = " << vet[i] << endl;
13. cout << "*(f + " << i << ") = " << *(f+i) << endl;
14. cout << "&vet[" << i << "] = " << &vet[i] << endl;
15. cout << "(f + " << i << ") = " << f+i << endl;
16. }
17. }
```

- 18 Assumindo que `pulo[]` é um vetor do tipo `int`, quais das seguintes expressões referenciam o valor do terceiro elemento do vetor?
- `*(pulo + 2);`
  - `*(pulo + 4);`
  - `pulo + 4;`
  - `pulo + 2;`
- 19 Considerando a declaração `int mat[4], *p, x;`, quais das seguintes expressões são válidas? Justifique.
- `p = mat + 1;`
  - `p = mat++;`
  - `p = ++mat;`
  - `x = (*mat)++;`
- 20 O que fazem os seguintes programas em C++?
- ```
int main(){
    int vet[] = {4,9,13};
    int i;
    for(i=0;i<3;i++){
        cout << *(vet+i) << " ";
    }
}

int main(){
    int vet[] = {4,9,13};
    int i;
    for(i=0;i<3;i++){
        cout << vet+i << " ";
    }
}
```
- 21 Seja `x` um vetor de 4 elementos, declarado da forma `TIPO x[4];`. Suponha que depois da declaração, `x` esteja armazenado no endereço de memória 4092 (ou seja, o endereço de `x[0]`). Suponha também que na máquina seja usada uma variável do tipo `char` ocupa 1 byte, do tipo `int` ocupa 4 bytes, do tipo `float` ocupa 6 bytes e do tipo `double` ocupa 8 bytes. Quais serão os valores de `x+1`, `x+2` e `x+3` se:
- (a) `x` for declarado como `char`?
 - (b) `x` for declarado como `int`?
 - (c) `x` for declarado como `float`?
 - (d) `x` for declarado como `double`?

Implemente um programa de computador para testar estas suposições e compare as respostas oferecidas pelo programa com as respostas que você idealizou.

- 22 Suponha que as seguintes declarações tenham sido realizadas:

```
#include <iostream>
using namespace std;
int main(){
    float aloha[10], coisas[10][5], *pf, value = 2.2;
    int i=3;
    aloha[2] = value;
    cout << &aloha << endl;
    aloha = value;
    cout << aloha << endl;
    coisas[4][4] = aloha[3];
    coisas[5] = aloha;
    pf = value;
    pf = aloha;
}
```

- 23 O que é um ponteiro para uma função? Pesquise na Internet referências sobre o assunto e escreva um pequeno programa exemplificando o uso deste recurso.