



Lista 1

comandos de decisão, comandos de repetição, funções, macros e vetores

1 Comandos de decisão

Questão 1.1:

Leia os quatro valores correspondentes aos eixos x e y de dois pontos quaisquer no plano, $p_1(x_1, y_1)$ e $p_2(x_2, y_2)$ e calcule a distância entre eles, mostrando 4 casas decimais após a vírgula, segundo a fórmula:

$$\text{Distancia} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Questão 1.2:

Escreva um programa em C para verificar se um determinado número de entrada é par ou ímpar.

Questão 1.3:

Qual será a saída do seguinte programa em C?

```
1 #include <stdio.h>
2 #include <math.h>
3 int main() {
4     float a = 0.0;
5     double b = 0.0;
6     a = sqrt(27.0);
7     b = sqrt(27.0);
8     if(a == b)
9         printf("Sao iguais");
10    else
11        printf("Sao diferentes");
12    return 0;
13 }
```

- a) Erro de compilação
- b) Sao iguais
- c) Sao diferentes
- d) Erro de execução

Questão 1.4:

Escreva um programa em C para encontrar o maior de três números dados por entrada.

Questão 1.5:

Qual será a saída do seguinte programa em C?

```
1 #include <stdio.h>
2 int main()
3 {
4     if (printf("0"))
5         printf("dentro do bloco if");
6     else
7         printf("dentro do bloco else");
8     return 0;
9 }
```

- a) dentro do bloco if
- b) dentro do bloco else
- c) 0dentro do bloco else
- d) 0dentro do bloco if

Questão 1.6:

Escreva um programa em C para converter dias específicos em anos, semanas e dias.

obs: Ignore o ano bissexto.

Questão 1.7:

Utilizando o comando 'switch', crie um programa no qual o usuário entre com uma letra (correspondendo a uma vitamina), por ex.: a,b,c,d... Em seguida, de acordo com a opção entrada, o programa deverá mostrar na tela informações sobre a vitamina de entrada.

Questão 1.8:

Escreva um programa em C para ler a temperatura em graus centígrados e exibir uma mensagem adequada de acordo com o estado da temperatura abaixo:

Temp < 0 então Tempo congelante
Temp 0-10, então, tempo muito frio
Temp 10-20, então, tempo frio
Temp 20-30, então, Normal
Temp 30-40, então, está quente
Temp >= 40, então, está muito quente

Questão 1.9:

Qual será a saída do seguinte programa em C?

```
1 #include <stdio.h>
2 int x = 0;
3 int main() {
4     if (x == x)
5         printf("Oi esse eh o if");
6     else
7         printf("Oi esse eh o else");
8     return 0;
9 }
```

- a) Oi esse eh o if
- b) Não mostra nada
- c) erro de compilação
- d) Oi esse eh o else

2 Comandos de repetição

Questão 2.1:

Escreva um programa em C para mostrar um número em ordem inversa.

Questão 2.2:

Escreva um programa em C para verificar se um número é um palíndromo ou não

Questão 2.3:

Escreva um programa em C para fazer tal padrão como uma pirâmide (n linhas) com números aumentados em 1.

ex. $n = 4$:

```
  1
 2 3
4 5 6
7 8 9 10
```

Questão 2.4:

Escreva um programa em C para fazer tal padrão como uma pirâmide (n linhas) com um número que irá repetir o número na mesma linha.

ex. $n = 4$:

```
  1
 2 2
3 3 3
4 4 4 4
```

Questão 2.5:

Escreva um programa em C que leia um inteiro x e exiba um triângulo de Pascal de tamanho x . Dados de teste:

Número de entrada de linhas: 5

Saída esperada:

```
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
```

Questão 2.6:

```
...
    for(i=1; i<=5; i++){
        for(j=1; j<i; j++){
            printf(" ");
        }
        for(j=i; j<=5; j++){
            printf("*");
        }
        printf("\n");
    }
...
```

Dado o programa em C acima, qual das seguintes alternativas mais representa o que será mostrado na tela?

a) *****

 **
 *

b) *****

 **
 *

c) *
 **

d) *

 **

e) Nenhuma das alternativas

Questão 2.7:

Escreva um programa em C para receber um valor n de entrada e “desenhar” na tela um triângulo retângulo com n linhas, usando asteriscos.

Exemplo de triângulo com $n = 4$

```
*
**
***
****
```

Questão 2.8:

Escreva um programa em C para receber um valor n de entrada e “desenhar” na tela um triângulo retângulo com n linhas, usando números. Cada linha será formada pelo respectivo número da linha.

Exemplo de triângulo com $n = 4$

```
1
22
333
4444
```

Questão 2.9:

Escreva um programa em C para receber um valor n de entrada e “desenhar” na tela um triângulo retângulo com n linhas, usando números. Cada posição será representada pelo valor da posição anterior + 1.

Exemplo de triângulo com $n = 4$

```
1
23
456
78910
```

Questão 2.10:

Escreva um programa em C para verificar se um caractere é uma letra, dígito ou caractere especial.

Questão 2.11:

Escreva um programa em C para exibir os primeiros 10 números naturais.

Questão 2.12:

Qual será a saída do seguinte programa em C?

```
1 #include<stdio.h>
2 int main(){
3     int i;
4     if(true)
5         printf("Isso funciona");
6     else
7         printf("Isso nao funciona");
8     return 0;
9 }
```

- a) Isso funciona
- b) Isso não funciona
- c) Erro de compilação
- d) Erro de execução

3 Macros e Funções

Questão 3.1:

Crie uma função recursiva $f(n)$ que retorne a soma da seguinte sequência numérica $n + (n - 1) + (n - 2) \dots 2$. No qual n é o primeiro termo e o número 2 é o último termo da sequência.

Ex. de entrada 1:
n=10

Saída:
soma = 54

Ex. de entrada 2:
n=5

Saída:
soma = 14

Questão 3.2:

O que é mostrado na execução do seguinte programa em C?

```
#include<stdio.h>
int main(){
    function();
    return 0;
}
void function(){
    printf("Aula de lp");
}
```

- a) Aula de lp
- b) Nada é mostrado
- c) Ocorre erro de execução
- d) Ocorre erro de compilação

Questão 3.3:

Codifique a função `fat(n)`, que devolve o fatorial de `n`.

Questão 3.4:

Codifique a função `pot(x,n)`, que devolve `x` elevado a `n`.

Questão 3.5:

Dada uma string, escreva uma função recursiva em C que conte o número total de consoantes nela. Uma consoante é um caractere do alfabeto que não é vogal (a, e, i, o e u). Exemplos de consoantes são b, c, d, f, g, ..

Questão 3.6:

Qual é a saída do seguinte programa em C?

```
int recursive(int i){
    int count = 0;
    count = count + i;
    return count; }
int main(){
    int i, j=0;
    for (i = 0; i <= 5; i++)
        j = recursive(i);
    printf("%d\n", j);
    return 0; }
```

- a) 5
- b) Erro de compilação
- c) 15
- d) 0

Questão 3.7:

Qual é a saída do seguinte programa em C?

```
int recursive(int i){
    static int count = 0;
    count = count + i;
    return count; }
int main(){
    int i, j=0;
    for (i = 0; i <= 5; i++)
        j = recursive(i);
    printf("%d\n", j);
    return 0; }
```

a) 5

b) Erro de compilação

c) 15

d) 0

Questão 3.8:

Qual é a saída do seguinte programa em C?

```
int x = -1;
int main(){
    while(x++ == 1)
        printf("loop ");
    return 0; }
```

a) loop

b) loop loop

c) loop loop loop

d) Nada é mostrado

Questão 3.9:

Qual será a saída do seguinte código em C?

```
1 #include <stdio.h>
2 int main()
3 {
4     int a = 1, b = 1, c;
5     c = a++ + b;
6     printf("a=%d, b=%d", a, b);
7 }
```

a) a = 1, b = 1

b) a = 2, b = 1

c) a = 1, b = 2

d) a = 2, b = 2

Questão 3.10:

Qual será a saída do seguinte código em C?

```
1 #include <stdio.h>
2 int main()
3 {
4     int a = 10, b = 10;
5     if (a = 5)
6         b--;
7     printf("%d, %d", a, b--);
8 }
```

a) a = 10, b = 9

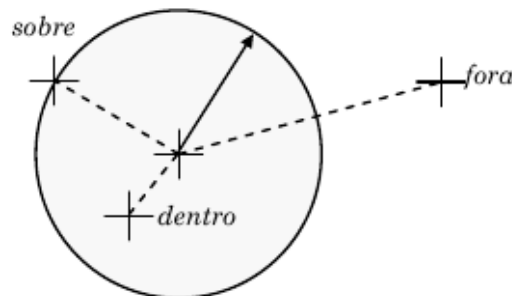
b) a = 10, b = 8

c) a = 5, b = 9

d) a = 5, b = 8

Questão 3.11:

São dadas as coordenadas (x_c, y_c) do centro de uma circunferência e a medida r de seu raio. Também são dadas as coordenadas (x, y) de uma série de pontos, sendo que o último deles é igual ao centro. Determine quantos pontos desta série estão dentro da circunferência, quantos estão fora e quantos estão sobre ela. Crie a função $dist(x_1, y_1, x_2, y_2)$ que dá a distância entre os pontos (x_1, y_1) e (x_2, y_2) e, depois, utilize-a no programa que resolva o problema proposto.

**Questão 3.12:**

Qual será a saída do seguinte programa em C?

```
#include<stdio.h>
#define x 2
int main()
{
    int i;
    i = x*x*x;
    printf("%d",i);
    return 0;
}
```

- a) 8
- b) x não foi declarado
- c) nenhuma saída
- d) lixo da memória

Questão 3.13:

Descreva o que a seguinte função em C faz.

```
int func(int x){
    if(x == 0)
        return 0;
    return ((x % 10) + func(x / 10));
}
```

Questão 3.14:

Escreva um programa em C para calcular o fatorial de um determinado número.

Questão 3.15:

Escreva um programa em C para exibir os n termos das séries harmônicas e sua soma. $1 + 1/2 + 1/3 + 1/4 + 1/5 \dots 1/n$ termos

Questão 3.16:

Escreva um programa em C para exibir os primeiros n termos da série Fibonacc.

Série de Fibonacci: 0 1 1 2 3 5 8 13

Questão 3.17:

Qual será a saída do seguinte programa em C?

```
#include<stdio.h>
int main(){
int i, j;
for(i = 1, j = 1; i<=3, j<=3; i++, j++)
    printf("%d %d ", i, j);
return 0;
}
```

- a) Erro de compilação
- b) 1 2 3 1 2 3
- c) 1 1 2 2 3 3
- d) nenhuma das alternativas acima

Questão 3.18:

Qual a saída produzida pela instrução a seguir?

```
printf("%d %d %d %d", !3, !0, 3+'a'>'b'+2 && !'b', 1 || !2 && 3);
```

Questão 3.19:

Escreva uma função recursiva em C para mostrar o n -ésimo termo da sequência fibonacci. n deve ser lido do teclado.

Sequência fibonacci: 0 1 1 2 3 5 8 13

Questão 3.20:

Para cada problema a seguir defina uma função recursiva, faça a simulação por substituição e desenhe o fluxo de chamadas e retornos:

- a) Calcular o fatorial de um número natural.
- b) Calcular o resto da divisão inteira usando subtração.
- c) Calcular o quociente da divisão inteira usando subtração.
- d) Calcular o produto de dois naturais usando adição.
- e) Calcular a soma de dois naturais usando as funções $\text{suc}(n)$ e $\text{pred}(n)$ que devolvem, respectivamente, o sucessor e o predecessor de um natural n .

Questão 3.21:

Defina os seguintes procedimentos recursivos ¹:

- a) regr(n), que exibe uma contagem regressiva a partir de n.
- b) bin(n), que exibe o número natural n em binário.

Questão 3.22:

Qual a saída do seguinte programa em C?

```
#include<stdio.h>
int main()
{
    int i = 5, j = 6, k = 7;
    if(i > j == k)
        printf("%d %d %d", i++, ++j, --k);
    else
        printf("%d %d %d", i, j, k);
    return 0;
}
```

- a) 5 7 6
- b) 5 6 7
- c) 6 6 6
- d) 5 7 7

Questão 3.23:

Qual será a saída do seguinte programa em C?

```
#include<stdio.h>
#define sqr(x) ++x * ++x
int main() {
    int a = 3, z;
    z = ++a * ++a;
    a -= 3;
    printf("%d %d", sqr(a), z);
    return 0; }
```

- a) 25 16
- b) 16 16
- c) 16 25
- d) 25 25

¹Procedimentos recursivos são definidos basicamente da mesma forma que as funções recursivas. A única diferença é que, como eles não devolvem resposta, não precisamos usar o comando return com as chamadas recursivas.

4 Vetores e matrizes

Questão 4.1:

Escreva um programa em C para ler 10 números do teclado e encontrar sua soma e média

Questão 4.2:

Escreva um programa em C para ler n número de valores em um array e exibi-lo na ordem inversa.

Dados de teste :

Insira o número de elementos a serem armazenados no array: 3

Insira 3 elementos no array:

elemento - 0: 2

elemento - 1: 5

elemento - 2: 7

Resultado esperado:

Os valores armazenados no array são:

2 5 7

Os valores armazenados no array ao contrário são:

7 5 2

Questão 4.3:

Codifique uma função que receba um número real n e devolva sua raiz quadrada r . Para calcular r , use o método proposto por Newton:

1. chuta-se um valor inicial para a raiz igual a 1
2. caso $|r^2 - n|$ seja inferior a 0.001, r é a resposta (fim)
3. caso contrário, aproxima-se $r = (r^2 + n)/(2r)$ e retorna-se ao 2º passo

Questão 4.4:

Qual será a saída do seguinte programa em C?

```
#include <stdio.h>
int main() {
    int a, b, c;
    int arr[5] = { 1, 2, 3, 25, 7 };
    a = ++arr[1];
    b = arr[1]++;
    c = arr[a++];
    printf("%d, %d e %d", a, b, c);
    return 0;
```

}

- a) 3, 3 e 25 b) 4, 4 e 25 c) 3, 4 e 25 d) 4, 3 e 25 e) n.d.a.

Questão 4.5:

Qual a diferença entre os dois seguintes códigos?

```
1 #include <stdio.h> //programa 1
2 int main()
3 {
4     int d, a = 1, b = 2;
5     d = a++ + ++b;
6     printf("%d %d %d", d, a, b);
7 }
```

```
1 #include <stdio.h> // programa 2
2 int main()
3 {
4     int d, a = 1, b = 2;
5     d = a++ +++b;
6     printf("%d %d %d", d, a, b);
7 }
```

- a) Nenhuma diferença, pois o espaço não faz diferença; os valores de a, b, d são iguais nos dois casos
- b) O espaço faz diferença, os valores de a, b, d são diferentes
- c) O programa 1 apresenta erro de sintaxe, o programa 2 não
- d) O programa 2 apresenta erro de sintaxe, o programa 1 não

Questão 4.6:

Escreva um programa em C para ler 10 números do teclado e encontre sua soma e média

Questão 4.7:

Usando vetores, codifique um programa para solicitar 5 números, via teclado, e exibí-los na ordem inversa àquela em que foram fornecidos.

Questão 4.8:

Escreva um programa em C para reorganizar um dado *array* na seguinte ordem: menor, maior, 2º menor, 2º maior, 3º menor, 3º maior e assim por diante.

Exemplo:

> O array de entrada é:

5 8 1 4 2 9 3 7 6

> O novo array é:

1 9 2 8 3 7 4 6 5

Questão 4.9:

Escreva um programa para encontrar o maior sub-vetor com número igual de 0s e 1s.

ex.:

vetor : 0 1 0 0 1 1 0 1 1 1

saída: sub-vetor encontrado do índice 0 ao 7

Questão 4.10:

Escreva um programa em C que subtraia dois vetores.

Questão 4.11:

Escreva um programa em C para encontrar a mediana de dois vetores ordenados de mesmo tamanho

Questão 4.12:

Escreva um programa em C para encontrar os dois elementos cuja soma é a mais próxima de zero.

ex.:

vetor: 38 44 63 -51 -35 19 84 -69 4 -46

par de elementos cuja soma é mínima: [44, -46]

Questão 4.13:

Escreva um programa em C para mover todos os zeros para o final de um determinado vetor.

Questão 4.14:

Os pares (749,400), (749 ,400), (841,400), (749,400), (1000,400), (844,800), (749,400), (749,400), (841,400), (749 ,400), (1122,400), (1000, 800), (749,400), (749,400), (1498,400), (1260, 400), (1000 ,400), (944,400), (841,800), (1335,400), (1335,400), (1260,400), (1000,400), (1122,400) e (1000,800) representam as frequências e durações de cada uma das notas de uma canção. Usando as funções `sound()`, `delay()` e `nosound()`, definidas em `conio.h`, crie um programa para tocá-la.

Questão 4.15:

Escreva um programa em C para encontrar o segundo menor elemento em uma matriz.

Questão 4.16:

Crie um programa em C para rotacionar à esquerda um vetor em n posições.

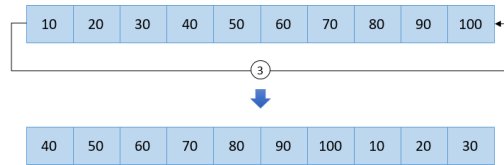


Figura 1: Rotação do vetor com $n = 3$.

Questão 4.17:

Crie um programa em C para contabilizar a quantidade de elementos repetidos em um vetor de inteiros, lido do teclado.

Questão 4.18:

Escreva uma função para verificar se os elementos de um vetor, formam um subconjunto de elementos de um segundo vetor.

Questão 4.19:

```
for(row=0; row<SIZE; row++){
    for(col=0; col<SIZE; col++){
        if(row+col == ((SIZE+1)-2)){
            x += A[row][col];
        }
    }
}
```

O que faz o programa em C descrito acima?

- a) encontra a soma dos elementos da diagonal principal de A
- b) encontra a soma dos elementos da matriz triangular superior de A
- c) encontra a soma dos elementos da matriz triangular inferior de A
- d) encontra a soma dos elementos da diagonal secundária de A

Questão 4.20:

O que é mostrado no seguinte programa em C?

```
for(i=0; i<n; i++){
    ctr=0;
    for(j=0,k=n; j<k+1; j++){
        if (i!=j){
            if(arr1[i]==arr1[j]){
                ctr++;
            }
        }
    }
    if(ctr==0){
        printf("%d ",arr1[i]);
    }
}
```

- a) Todos os elementos únicos em arr1
- b) Todos os elementos duplicados em arr1
- c) A frequência de cada elemento em arr1
- d) Número de elementos fora da coluna principal

Questão 4.21:

Qual será a saída do seguinte programa em C?

```
#include<stdio.h>
int main(){
    int rows = 3, columns = 4, i, j, k;
    int a[3][4] = {1, 2, 3, 5, 7};
    i = j = k = 00;
    for(i = 0; i<rows; i++)
    for(j = 0; j<columns; j++)
    if(a[k][j]<k)
    k = a[i][j];
    printf("%d\n", k);
    return 0;
}
```

- a) 00
- b) nada é mostrado
- c) 0
- d) 7

Questão 4.22:

Qual será a saída do seguinte programa em C?

```
#include<stdio.h>
int main()
{
    int arr[2] = {1, 2, 3, 4, 5};
    printf("%d", arr[3]);
    return 0;
}
```

- a) 3
- b) 4
- c) Algum lixo da memória
- d) Erro de compilação

Questão 4.23:

Qual será a saída do seguinte programa em C?

```
#include<stdio.h>
int main()
{
    int a, b, c;
    int arr[5] = {1, 2, 3, 25, 7};
    a = ++arr[1];
    b = arr[1]++;
    c = arr[a++];
    printf("%d--%d--%d", a, b, c);
    return 0;
}
```

- a) 4-3-25
- b) 3-3-25
- c) 4-4-25
- d) 3-4-25

Desafios ☺

Problema do investimento

Imagine que você é um investidor na bolsa de valores e tem X reais em caixa, disponíveis para o investimento. Na sua carteira de ações, você possui n ações disponíveis para investimento. Cada ação i possui um retorno financeiro l_i e custo c_i , caso você decida investir nela.

Considere que ou você investe na ação ou não investe. Não é permitido investir uma porcentagem da ação. Por exemplo, se uma ação 1 possui $l_1 = 10$ e $c_1 = 5$, e você decide investir nela; o seu retorno financeiro com essa ação será de 10 reais e o custo será de 5 reais.

Crie um programa em C que, dado X reais em caixa e n ações com seus respectivos custos e retornos, encontre o melhor investimento possível, maximizando o retorno financeiro total. Deverão ser mostrados na tela, quais ações foram escolhidas e o valor do retorno financeiro total atingido.

Obs. 1: Lembre que você tem apenas X reais para investir nas ações, portanto, o custo total do investimento não pode ultrapassar X .

Obs. 2: Esta questão será pontuada proporcionalmente ao retorno financeiro obtido. Quanto maior o retorno financeiro, mais próximo da pontuação máxima.

Exemplo de entrada

`x=77`

`n=7`

`Lucros[7]={70, 20, 29, 37, 7, 5, 10}`

`Custos[7]={31, 21, 20, 19, 4, 3, 6 }`

Possível saída:

As ações escolhidas foram: 0, 3, 4, 5 e 6

Lucro total = 129.00

Problema do entregador de pizza

Um entregador de pizza deve fazer entrega em n casas de uma cidade. Dado uma distância c_{ij} para ir de uma casa i a uma casa j , o entregador deve visitar todas as casas, exatamente uma vez cada uma, e voltar para o ponto de origem.

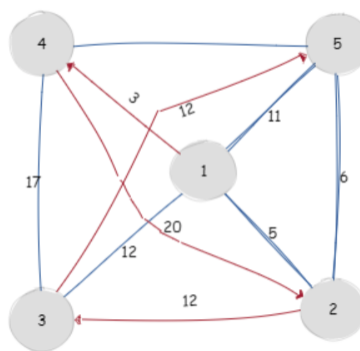


Figura 2: Exemplo de rota do entregador de pizza, partindo da pizzaria (ponto 1)

Crie um programa em C que, dado um “mapa” (matriz de distâncias $n \times n$), mostre na tela uma possível rota para o entregador, de forma que o percurso seja o menor possível. Por fim, apresente a rota por meio de uma função recursiva.

Ex. de matriz no código:

```
int distancias[5][5] = {{-1, 5, 12, 3, 11},
                        {5, -1, 12, 20, 6},
                        {12, 12, -1, 17, 12},
                        {3, 20, 17, -1, 20},
                        {11, 6, 12, 20, -1}};
```

Possível saída (em tela):

Rota: 0 -> 3 -> 1 -> 2 -> 4 -> 0

Custo da rota:

$3 + 20 + 12 + 12 + 11 = 58$

Ex. de protótipo de função:

```
void printf_recursoivo(int n_casas, int rota[]);
```