roteiro-01/ex01-01.c

```
#include <stdio.h>
   #define TAMANHO 10
 3
 5
   int negativos (float* vet, int N) {
 6
        int total = 0;
 7
        for (int i = 0; i < N; i++) {
 8
            if (vet[i] < 0) total++;</pre>
 9
10
        return total;
   }
11
12
13
   int main () {
14
        float vetor[TAMANH0] = { 10, -9, 5, -1, -2, 4, 3, 0, 6, -7 };
15
16
        printf("[");
17
        for (int i = 0; i < TAMANHO; i++) {</pre>
            printf("%.2f", vetor[i]);
18
19
            if (i < TAMANHO - 1) printf(", ");</pre>
20
21
        printf("]\n");
22
       printf("Quantidade de negativos = %d\n", negativos(vetor, TAMANHO));
23
24 }
25
```





≡







gabriel-dp@gabriel-dp:~/Desktop/dev/c/lab2/roteiro-01\$./ex01-01
[10.00, -9.00, 5.00, -1.00, -2.00, 4.00, 3.00, 0.00, 6.00, -7.00]
Quantidade de negativos = 4
gabriel-dp@gabriel-dp:~/Desktop/dev/c/lab2/roteiro-01\$

roteiro-01/ex01-02.c

```
#include <stdio.h>
   #include <stdlib.h>
   #include <time.h>
   #define TAMANHO 10
5
6
   void preenche(double* vet, int n) {
7
        srand(time(NULL));
8
        for (int i = 0; i < n; i++) {
9
            vet[i] = (rand() % 200) - 100;
10
        }
11
   }
12
13
   double max(double* vet, int n) {
14
        double maior = vet[0];
15
        for (int i = 1; i < n; i++) {
16
            if (vet[i] > maior) maior = vet[i];
17
        }
18
        return maior;
19
   }
20
21
   double min(double* vet, int n) {
22
        double menor = vet[0];
23
        for (int i = 1; i < n; i++) {
24
            if (vet[i] < menor) menor = vet[i];</pre>
25
        }
26
        return menor;
27
28
29
   double media(double* vet, int n) {
30
        double total = 0;
31
        for (int i = 0; i < n; i++) {
            total += vet[i];
32
33
34
        return (total / n);
35
   }
36
37
   int main() {
38
        double vetor[TAMANH0];
39
        preenche(vetor, TAMANHO);
40
41
        printf("[");
42
        for (int i = 0; i < TAMANHO; i++) {
43
            printf("%.0f ", vetor[i]);
44
            if (i < TAMANHO - 1) printf(", ");</pre>
45
        printf("]\n");
46
47
48
        printf("min = %.0f\n", min(vetor, TAMANHO));
49
        printf("max = %.0f\n", max(vetor, TAMANHO));
50
        printf("media = %.2f\n", media(vetor, TAMANHO));
51
52
        return 0;
53
   }
54
```



Q = -







```
gabriel-dp@gabriel-dp:~/Desktop/dev/c/lab2/roteiro-01$ ./ex01-02
[3 , 50 , -66 , -100 , 59 , 83 , -80 , 27 , 90 , 68 ]
min = -100
max = 90
media = 13.40
```

roteiro-01/ex01-03.c

```
#include <stdio.h>
 2
   #include <string.h>
 3
 4
   typedef struct {
5
        char nome[20];
6
        int matricula;
7
        double nota;
   } Aluno;
10
   void imprimeAluno(Aluno aluno) {
        printf("Nome = %s | Matricula = %d | Nota = %.2f\n", aluno.nome,
11
    aluno.matricula, aluno.nota);
12
13
   void imprimeMax(Aluno* alunos, int n) {
14
15
        int max = 0;
16
        for (int i = 1; i < n; i++) {</pre>
17
            if (alunos[i].nota > alunos[max].nota) max = i;
18
        }
19
        printf("Maior - ");
20
        imprimeAluno(alunos[max]);
21
   }
22
23
   void imprimeMin(Aluno* alunos, int n) {
24
        int min = 0;
25
        for (int i = 1; i < n; i++) {
            if (alunos[i].nota < alunos[min].nota) min = i;</pre>
26
27
28
        printf("Menor - ");
29
        imprimeAluno(alunos[min]);
30
   }
31
   int main() {
32
33
        Aluno a, b, c;
        strcpy(a.nome, "Ana");
34
35
        a.matricula = 1;
36
        a.nota = 10;
        strcpy(b.nome, "Davi");
37
38
        b.matricula = 2;
39
        b.nota = 9.9;
        strcpy(c.nome, "Prenassi");
40
        c.matricula = 3;
41
42
        c.nota = 9.8;
43
        Aluno alunos[] = {a, b, c};
44
45
        imprimeMax(alunos, sizeof(alunos) / sizeof(alunos[0]));
46
        imprimeMin(alunos, sizeof(alunos) / sizeof(alunos[0]));
47
48
        return 0;
49 }
```











```
gabriel-dp@gabriel-dp:~/Desktop/dev/c/lab2/roteiro-01$ ./ex01-03
```

Maior - Nome = Ana | Matricula = 1 | Nota = 10.00

Menor - Nome = Prenassi | Matricula = 3 | Nota = 9.80

roteiro-01/ex01-04.c

```
#include <math.h>
 2
   #include <stdio.h>
 3
    int raizes(float a, float b, float c, float* x1, float* x2) {
 5
        float delta = ((b * b) - (4 * a * c));
 6
 7
        if (delta < 0) return 0;</pre>
 8
        if (delta == 0) {
 9
10
            *x1 = ((-b) / (2 * a));
11
            return 1;
12
        }
13
        *x1 = ((-b + sqrt(delta)) / (2 * a));
14
        *x2 = ((-b - sqrt(delta)) / (2 * a));
15
16
        return 2;
17
   }
18
19
   int main() {
20
        float a, b, c, x1, x2;
21
        printf("ax^2: ");
22
        scanf("%f", &a);
23
        printf("bx: ");
24
        scanf("%f", &b);
25
        printf("c: ");
26
        scanf("%f", &c);
27
28
        switch (raizes(a, b, c, &x1, &x2)) {
29
            case 1:
30
                printf("uma raiz real\n");
                printf("x1 = %f", x1);
31
32
                break;
33
            case 2:
34
                printf("duas raizes reais\n");
35
                printf("x1 = %f | x2 = %f", x1, x2);
36
                break;
37
            default:
38
                printf("nao existe raiz real\n");
39
                break;
40
        printf("\n");
41
42
43
        return 0;
44 }
```





Q =





```
gabriel-dp@gabriel-dp:~/Desktop/dev/c/lab2/roteiro-01$ ./ex01-04
```

```
ax^2: 2
bx: -5
c: -7
```

duas raizes reais

 $x1 = 3.500000 \mid x2 = -1.000000$

roteiro-01/ex02-01.c

```
#include <stdio.h>
 3
   void printCrescente(int i, int max) {
        if (i > max) return;
        printf("(%d)", i);
 5
 6
        printCrescente(i + 1, max);
 7
   }
   void printDecrescente(int min, int i) {
 9
10
        if (i < min) return;</pre>
11
        printf("(%d)", i);
        printDecrescente(min, i - 1);
12
13
   }
14
15 int main () {
16
        printCrescente(1, 5);
17
        printf("\n");
        printDecrescente(1, 5);
18
19
        printf("\n");
20
        return 0;
21
22 }
```



```
gabriel-dp@gabriel-dp:~/Desktop/dev/c/lab2/roteiro-01$ ./ex02-01
(1)(2)(3)(4)(5)
(5)(4)(3)(2)(1)
gabriel-dp@gabriel-dp:~/Desktop/dev/c/lab2/roteiro-01$
```

roteiro-01/ex02-02.c

```
1 #include <stdio.h>
   #include <string.h>
 3
   #include <ctype.h>
 5
   #define MAX CHAR 255
 6
 7
   void imprimeLetrasDecrescente(char* vetor, int atual) {
 8
        if (atual > MAX CHAR) {
            printf("\n");
 9
10
            return;
11
        }
12
        int length = strlen(vetor);
13
14
        for (int i = 0; i < length; i++) {</pre>
            if (vetor[i] == (char)(MAX_CHAR - atual)) printf("%c", (char)(MAX_CHAR -
15
   atual));
16
        }
17
18
        imprimeLetrasDecrescente(vetor, atual + 1);
19
   }
20
21
   int main () {
22
        imprimeLetrasDecrescente("gabriel", 0);
23
24
        return 0;
25 }
```





≡



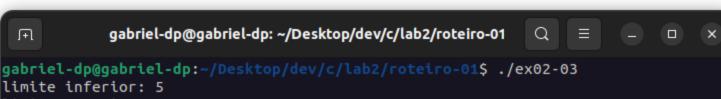




gabriel-dp@gabriel-dp:~/Desktop/dev/c/lab2/roteiro-01\$./ex02-02
rligeba
gabriel-dp@gabriel-dp:~/Desktop/dev/c/lab2/roteiro-01\$

roteiro-01/ex02-03.c

```
1 #include <stdio.h>
   int somaIntervaloRecursivo(int atual, int inf, int sup) {
 3
        if (atual < inf) return 0;</pre>
        return atual + somaIntervaloRecursivo(atual - 1, inf, sup);
 6
   }
 7
   int main() {
 9
        int inf, sup, soma = 0;
10
       printf("limite inferior: ");
11
        scanf("%d", &inf);
12
        printf("limite superior: ");
13
14
        scanf("%d", &sup);
        printf("somatorio = %d\n", somaIntervaloRecursivo(sup, inf, sup));
15
16
17
       return 0;
18 }
```



limite inferior: 5
limite superior: 10
somatorio = 45
gabriel-dp@gabriel-dp:~/Desktop/dev/c/lab2/roteiro-01\$

roteiro-01/ex02-04.c

```
#include <stdio.h>
 3
   int multiplica(int m, int n) {
 4
        if (n == 0) return 0;
 5
        return m + multiplica(m, n - 1);
6
   }
 7
   int main () {
 9
        int n, m;
10
        printf("Primeiro numero: ");
11
        scanf("%d", &n);
       printf("Segundo numero: ");
12
13
        scanf("%d", &m);
14
       printf("Produto = %d\n", multiplica(n, m));
15
16
        return 0;
17 }
```

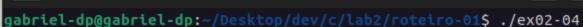










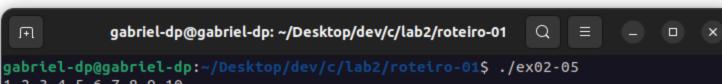


Primeiro numero: 10 Segundo numero: 20

Produto = 200

roteiro-01/ex02-05.c

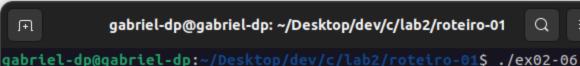
```
#include <stdio.h>
 3
   #define N 10
 5
    void preencheRecursivo(int i, int n, int* vetor) {
 6
        if (i >= n) return;
        vetor[i] = i + 1;
 7
 8
        preencheRecursivo(i + 1, n, vetor);
 9
10
   void imprimeVetorCrescente(int i, int n, int* vetor) {
11
        if (i >= n) return;
12
        printf("%d ", vetor[i]);
13
14
        imprimeVetorCrescente(i + 1, n, vetor);
15
   }
16
17
   void imprimeVetorDecrescente(int i, int n, int* vetor) {
        if (i >= n) return;
18
19
        printf("%d ", vetor[n - 1 - i]);
20
        imprimeVetorDecrescente(i + 1, n, vetor);
21
22
23
   int maiorRecursivo(int i, int n, int* vetor) {
        int next = i < n - 1 ? maiorRecursivo(i + 1, n, vetor) : vetor[i];</pre>
24
25
        return (vetor[i] > next) ? vetor[i] : next;
26
   }
27
   int main() {
28
29
        int vetor[N];
30
        preencheRecursivo(0, N, vetor);
31
        imprimeVetorCrescente(0, N, vetor);
32
        printf("\n");
33
        imprimeVetorDecrescente(0, N, vetor);
34
        printf("\n");
35
        printf("maior = %d\n", maiorRecursivo(0, N, vetor));
36
37
        return 0;
38 }
39
```



```
gabriel-dp@gabriel-dp:~/Desktop/dev/c/lab2/rotelro-01$ ./ex02-05
1 2 3 4 5 6 7 8 9 10
10 9 8 7 6 5 4 3 2 1
maior = 10
gabriel-dp@gabriel-dp:~/Desktop/dev/c/lab2/rotelro-01$
```

roteiro-01/ex02-06.c

```
#include <stdio.h>
3
   void rec_hanoi (int n, char origem, char dest, char trab) {
4
      if (n > 0) {
5
          rec_hanoi(n - 1, origem, trab, dest);
6
          printf("Mova o disco %d para o pino %c para %c\n", n, origem , dest);
7
          rec hanoi(n - 1, trab, dest, origem);
8
      }
9
   }
10
   int main () {
11
12
      int discos;
      printf("Digite a quantidade de discos:\n");
13
14
      scanf("%d", &discos);
15
      printf("----\n");
16
      17
18
19
      return 0;
20
21 }
```



```
Q ≡ – □
x02-06
```

```
Digite a quantidade de discos: 4
            -RECURSIVO-
Mova o disco 1 para o pino A para B
Mova o disco 2 para o pino A para C
Mova o disco 1 para o pino B para C
Mova o disco 3 para o pino A para B
Mova o disco 1 para o pino C para A
Mova o disco 2 para o pino C para B
Mova o disco 1 para o pino A para B
Mova o disco 4 para o pino A para C
Mova o disco 1 para o pino B para C
Mova o disco 2 para o pino B para A
Mova o disco 1 para o pino C para A
Mova o disco 3 para o pino B para C
Mova o disco 1 para o pino A para B
Mova o disco 2 para o pino A para C
Mova o disco 1 para o pino B para C
```