# Link para acesso dos exercícios no GITHUB:

# Questão 1) Para cada exemplo apresentado, elabore o programa-teste

// Criei só 1 programa que chama ambas as funções.

#include <stdio.h>

#include <locale.h>

#include <stdlib.h>

#include <locale.h>

#define TAMANHO\_VET 5

float soma (float \* vetor);

float maior (float \* vetor);

float main()

{

setlocale (LC\_ALL, "Portuguese");

int i;

float vetor[TAMANHO\_VET], res\_soma, res\_maior;

for (i = 0; i < TAMANHO\_VET ; i++)

{

printf("Informe o valor da posição %d do vetor: \n", i+1);

scanf("%f", &vetor[i]);

}

res\_soma = soma(vetor);

printf("\n\nA soma é %.2f \n\n", res\_soma);

res\_maior = maior(vetor);

printf("\n\nO maior é %.2f \n\n", res\_maior);

}

*float soma (float \* vetor)*

*{*

*setlocale (LC\_ALL, "Portuguese");*

*int i;*

*float res=0;*

*printf ("\nres= %.2f\ttamanho\_vet = %d", res, TAMANHO\_VET);*

*for (i = 0; i < TAMANHO\_VET; i++)*

*{*

*res += vetor[i];*

*printf("\nres= %.2f\ttamanho\_vet = %d", res, TAMANHO\_VET);*

*}*

*return res;*

*}*

*float maior (float \* vetor)*

*{*

*setlocale (LC\_ALL, "Portuguese");*

*int i;*

*float res = vetor[0];*

*printf("\nmaior= %.2f\ttamanho\_vet = %i", res, TAMANHO\_VET);*

*for (i = 1; i < TAMANHO\_VET; i++)*

*{*

*if(vetor[i] > res)*

*res=vetor[i];*

*printf("\nmaior= %.2f\ttamanho\_vet = %i", res, TAMANHO\_VET);*

*}*

*return res;*

*}*

# Questão 2) Elabore o algoritmo que calcule a soma dos números pares de um vetor.

*int somapares (int \* vetor);*

*int main()*

*{*

*setlocale (LC\_ALL, "Portuguese");*

*int i, vetor[TAMANHO\_VET], res\_soma;*

*for (i = 0; i < TAMANHO\_VET ; i++)*

*{*

*printf("Informe o valor da posição %d do vetor: \n", i+1);*

*scanf("%i", &vetor[i]);*

*}*

*res\_soma = somapares(vetor);*

*printf("\n\nA soma é %i \n\n", res\_soma);*

*}*

*int somapares (int \*vetor)*

*{*

*int i, res=0;*

*for (i = 0; i < TAMANHO\_VET; i++)*

*{*

*if (vetor[i] % 2 == 0)*

*res += vetor[i];*

*}*

*return res;*

*}*

# Questão 3) Determine a complexidade de tempo para o algoritmo elaborado no item 2.

*int somapares (int \* vetor); //1*

*int main() //1*

*{*

*setlocale (LC\_ALL, "Portuguese"); //1*

*int i, vetor[TAMANHO\_VET], res\_soma; //1*

*for (i = 0; i < TAMANHO\_VET ; i++) //n+1*

*{*

*printf("Informe o valor da posição %d do vetor: \n", i+1); //n*

*scanf("%i", &vetor[i]); //n*

*}*

*res\_soma = somapares(vetor); //1*

*printf("\n\nA soma é %i \n\n", res\_soma); //1*

*}*

*int somapares (int \*vetor)*

*{*

*int i, res=0; //1*

*for (i = 0; i < TAMANHO\_VET; i++) //n+1*

*{*

*if (vetor[i] % 2 == 0) //n*

*res += vetor[i]; //n*

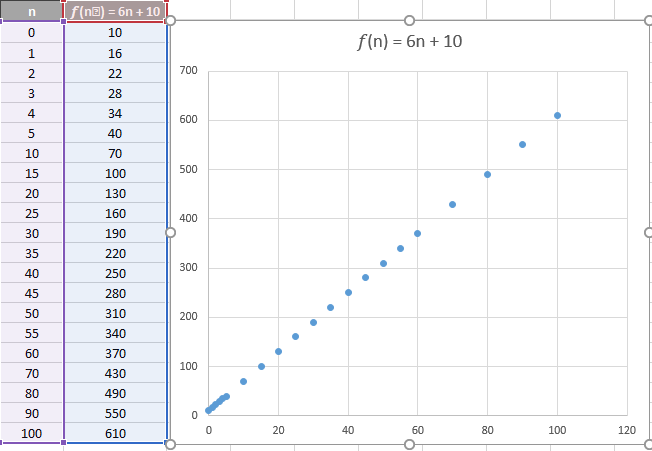
*}*

*return res; //1*

*}*

*//10 + 6n*

*A complexidade de Tempo dependerá do valor de N. No algoritmo em questão foi definido TAMANHO\_VET como 5, logo isso seria*



# Questão 4) Considerando o algoritmo do bubble\_sort, elabore o programa para testar esse algoritmo

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

void bubble\_sort(float \* vetor, int TAMANHO\_VETOR);

void mostra\_notas(float \* vetor, int TAMANHO\_VETOR);

int main()

{

setlocale (LC\_ALL, "Portuguese");

int i, TAMANHO\_VETOR;

float \*vetor;

printf("Informe o numero de componentes do vetor\n");

scanf("%d", &TAMANHO\_VETOR);

vetor = (float \*) malloc (TAMANHO\_VETOR \* sizeof(float));

for (i = 0; i < TAMANHO\_VETOR ; i++) {

printf("Informe o valor da posição %d do vetor: \n", i+1); //n

scanf("%f", &vetor[i]); //n

}

bubble\_sort(vetor, TAMANHO\_VETOR);

}

void bubble\_sort(float \* vetor, int TAMANHO\_VETOR)

{

int i, j;

float aux;

for (j = 0; j < TAMANHO\_VETOR-1; j++)

{

for (i = 0; i < TAMANHO\_VETOR-1; i++)

{

printf ("\nComparando %.2f com %.2f ", vetor[i], vetor[i+1]);

if (vetor [i] > vetor[i+1])

{

printf ("-> empurra %.2f para o fundo");

printf ("-> troca com %.2f", vetor[i], vetor[i+1]);

aux = vetor[i];

vetor[i] = vetor[i+1];

vetor [i+1] = aux;

}

if (j < TAMANHO\_VETOR-1)

mostra\_notas(vetor, TAMANHO\_VETOR);

}

}

}

void mostra\_notas(float \*vetor, int TAMANHO\_VETOR)

{

int i;

for (i = 0; i < TAMANHO\_VETOR ; i++) //n+1

{

printf("\nOrdenado por Bubble\_sort: Posição %i: \t %.2f\n", i+1, vetor[i]); //n

}

}