

## Instruções

Os problemas marcados com ▷ são os exercícios selecionados e que devem ser entregues no prazo determinado.

Prazo dos exercícios selecionados: 28/11/2021

## Lista 4 (semana 6)

1. Qual a saída do programa a seguir? (execute apenas para conferir)

```
#include <stdio.h>

void funcao0(int x, int y) {
    printf("%d %d\n", x, y);
}

void funcao1(int x, int y) {
    funcao0(x, y+1);
    printf("--\n");
    funcao0(x+1, y);
}

void funcao2(int x, int y) {
    funcao1(2*x, 2*y);
}

int main() {

    funcao2(1, 2);
    printf("\n");
    funcao2(7, 6);
    return 0;
}
```

2. Implemente as seguintes funções. Linhas em branco a seguir separam as questões (começa em q2a.c, depois vem q2b.c, etc). Siga o input/output especificado em cada bloco para implementar a função main.

```
//a) retorna o resultado do somatório de 1 até n
int somatorio(int n);
//input: n
//output: resultado do somatório de 1 até n

//b) retorna a próxima letra
//se a letra passada for z, deve retornar a (e para Z deve retornar A)
char proximaLetra(char letra);
//input: um caractere
//output: a próxima letra

//c) retorna um dos seguintes valores possíveis:
//0 - letra minúscula
//1 - letra maiúscula
//2 - número
int tipoDoCaractere(char caractere);
//input: um caractere
//output: letra minuscula ou letra maiuscula ou numero, a depender do caso
```

```

//d) retorna se x é primo
int ehPrimo(int x);
//retorna o número de primos no intervalo [a, b]
int numeroPrimos(int a, int b);
//input: a e b
//output: a quantidade de primos no intervalo [a, b]

//e) nada retorna, apenas escreve na tela a data por extenso
void escreverData(int dia, int mes, int ano);
//por exemplo, ao chamar escreverData(24, 11, 2021) será escrito
//...na tela "24 de novembro de 2021"
//input: dia, mes e ano
//output: a mensagem (escrita pela função)

```

3. Implemente as seguintes funções para resolver novamente as referidas questões de listas passadas:

```

//a) retorna a quantidade de dobras necessarias (questão 9 da lista 3)
int dobras(float comprimentoFolha, float larguraFolha, float comprimentoBolso);
//utilize essa função para resolver a questão 9 da lista 3

//b) retorna a soma dos divisores próprios
int somaDivisoresProprios(int x);
//utilize essa função para resolver a questão 10 da lista 3

```

4. ▷ Cansados das mesmas regras do jogo de dardos, um grupo de amigos resolveu inovar: ganha pontos não só quem acerta mais perto do alvo, mas ganha também uma pontuação bônus (metade da principal) quem acerta próximo do lançamento anterior. Afinal, quem lança errado sempre no mesmo lugar também tem boa pontaria, ela só não está calibrada.

O seu programa deve ler as coordenadas (x, y), dois números reais, de 10 lançamentos e escrever na tela a pontuação total.

Distância	Pontuação principal	Pontuação bônus
[0,1]	+10 pts	+5 pts
]1,2]	+6 pts	+3 pts
]2,3]	+4 pts	+2 pts
Acima de 3	0 pts	0 pts

A pontuação principal é atribuída em função da distância da coordenada de um lançamento e o centro do alvo (0, 0). A pontuação bônus é atribuída em função da distância da coordenada de um lançamento e a coordenada do lançamento anterior (não há pontuação bônus para o primeiro lançamento pois não há lançamento anterior).

Por exemplo, suponha que o terceiro lançamento tenha caído em (3, 4) e o lançamento anterior (o segundo) tenha sido em (2, 3). Então a distância para o centro do alvo é  $\sqrt{(3-0)^2 + (4-0)^2} = 5$  e a distância para o lançamento anterior é dado por  $\sqrt{(3-2)^2 + (4-3)^2} = \sqrt{2}$ . Verificando a tabela, nota-se que a pontuação principal é 0 pts ( $5 > 3$ ) e a pontuação bônus é 3 pts ( $\sqrt{2} \in ]1, 2[$ ). Logo são 0+3 pts pelo terceiro lançamento.

Sugestões de funções:

- uma função para calcular distância euclidiana entre dois pontos
- uma função para calcular a pontuação de um ponto em relação a outro

5. O filho de um amigo seu tem só 12 anos e está encantado com os números primos. Ele gosta tanto

que está montando uma lista com todos os trios de primos na forma  $(x, x + 2, x + 6)$ . Ele conseguiu encontrar o trio 641, 643, 647, mas levou quase uma tarde toda para encontrá-lo. Ele soube que você está aprendendo programação e é capaz de escrever um programa que escreva todos os trios de primos até 50000 quase instaneamente! É sua hora então de automatizar essa árdua tarefa usando a linguagem C.

A saída do programa deve ser todos os trios de primos na forma  $(x, x + 2, x + 6)$  até 50000, um por linha, no formato (x, x+2, x+6). Ou seja, as 4 primeiras linhas da saída devem ser:

```
(5, 7, 11)
(11, 13, 17)
(17, 19, 23)
(41, 43, 47)
```

Sugestões de funções:

- a. uma função para determinar de um número é primo ou não

6. Na seguinte implementação há duas funções além da main: funcao0 e funcao1. Na função main há várias chamadas a ambas funções. Como utilizar variáveis globais para verificar o número de chamadas a funcao0?

```
#include <stdio.h>

void funcao0(int x, int y) {
    printf("%d %d\n", x, y);
}

void funcao1(int x, int y) {
    funcao0(x, y+1);
    printf("--\n");
    funcao0(x+1, y);
}

int main() {

    //varias chamadas a funcao0 e a funcao1
    //escrever o total de chamadas a funcao0
    return 0;
}
```