**Atividades do conteúdo: Unidade 1 (Conceitos Básicos)**

**1 - 1** Declare e inicialize uma instância para cada um dos seguintes tipos de dados:

1. Smartphone, com o nome do modelo, memória e dimensões de tela.
2. Jogo de futebol, com os nomes dos times e o placar (faça o placar ser também um tipo de dado);

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  /\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/  typedef struct Smartphone{  char model[30];  int memory;  float height, width;  }TSmartphone;  typedef struct Scoreboard{  int teamLeft;  int teamRight;  }TScoreboard;  typedef struct SoccerMatch{  char nameTeamLeft[30];  char nameTeamRight[30];  TScoreboard \*scoreboard;  }TSoccerMatch;  int main(int argc, char \*argv[]) {  TSmartphone \*pSmartphone;  pSmartphone = (TSmartphone \*)malloc(sizeof(TSmartphone));    TSoccerMatch \*pSoccerMatch;  pSoccerMatch = (TSoccerMatch \*)malloc(sizeof(TSoccerMatch));    pSoccerMatch -> scoreboard = (TScoreboard \*)malloc(sizeof(TScoreboard));    printf("Digite o modelo: \n");  gets(pSmartphone -> model);    printf("Digite a memory: \n");  scanf("%i",&pSmartphone -> memory);    printf("Digite a altura e em seguida a largura: \n");  scanf("%f" "%f", &pSmartphone -> height, &pSmartphone -> width);    printf("\n%s \nMemoria de %iG\nAltura de %f\nLargura de %f \n", pSmartphone -> model, pSmartphone -> memory, pSmartphone -> height, pSmartphone -> width);  printf("\n=================================================================================================================\n\n");    fflush(stdin);    printf("Digite o nome do time A: \n");  gets(pSoccerMatch -> nameTeamLeft);    printf("Digite o nome do time B: \n");  gets(pSoccerMatch -> nameTeamRight);    printf("Placar do time A: \n");  scanf("%i",&pSoccerMatch -> scoreboard -> teamLeft);    printf("Placar do time B: \n");  scanf("%i",&pSoccerMatch -> scoreboard -> teamRight);    printf("\nTime %s: %i \nTime %s: %i", pSoccerMatch -> nameTeamLeft, pSoccerMatch -> scoreboard -> teamLeft,  pSoccerMatch -> nameTeamRight, pSoccerMatch -> scoreboard -> teamRight);    return 0;  } |

**1 - 2** Implemente o TAD TRegistroCorrida, conforme descrito na atividade 1-2.

|  |
| --- |
| #ifndef REGISTROCORRIDA\_H\_INCLUDED  #define REGISTROCORRIDA\_H\_INCLUDED  typedef struct RegistrosCorrida TRegistroCorrida;  TRegistroCorrida \*criar(int quantidade);  float calcularMedia(TRegistroCorrida \*registroCorrida, int quantidadeParaCalculo);  float maiorFrequencia(TRegistroCorrida \*registroCorrida, int quantidadeParaVetor);  void findByType(TRegistroCorrida \*registroCorrida, int quantidadeParaVetor, int defineRetorno);  #endif // REGISTROCORRIDA\_H\_INCLUDED  #include <stdio.h>  #include <stdlib.h>  #includ "registroCorrida.h"  struct RegistrosCorrida {  float frequencia;  float temperatura;  float elevacao;  };  TRegistroCorrida \*criar(int quantidade){  TRegistroCorrida \*registroCorrida = (TRegistroCorrida \*) malloc(quantidade\*sizeof(TRegistroCorrida));  return registroCorrida;  };  float calcularMedia(TRegistroCorrida \*registroCorrida, int quantidadeParaCalculo){  int i; float soma = 0;  for(i = 0; i < quantidadeParaCalculo; i++){  soma = soma + registroCorrida[i].frequencia;  }  return soma / quantidadeParaCalculo;  }  float maiorFrequencia(TRegistroCorrida \*registroCorrida, int quantidadeParaVetor){  int i; float maiorValor = 0;  for(i = 0; i < quantidadeParaVetor; i++){  if(registroCorrida[i].frequencia > maiorValor){  maiorValor = registroCorrida[i].frequencia;  }  }  return maiorValor;  }  void findByType(TRegistroCorrida \*registroCorrida, int quantidadeParaVetor, int defineRetorno){  int i;  switch(defineRetorno){  case 1:  printf("Lista de FREQUENCIAS");  for(i = 0; i < quantidadeParaVetor; i++){  printf("\nFrequencia: %f", registroCorrida[i].frequencia);  }break;  case 2:  printf("Lista de TEMPERATURAS");  for(i = 0; i < quantidadeParaVetor; i++){  printf("\nTemperatura: %f", registroCorrida[i].temperatura);  }break;  case 3:  printf("Lista de ELECACOES");  for(i = 0; i < quantidadeParaVetor; i++){  printf("\nElevacao: %f", registroCorrida[i].elevacao);  }break;  }  } |

**1 - 3** Implemente o TAD Jogo de Futebol com as seguintes funcionalidades:

1. Criar um jogo, que recebe dos dados de um jogo, inicializa uma variável Jogo de Futebol com esses dados, e retorna o ponteiro para ela;
2. Vencedor, que recebe um jogo e mostra na tela o nome do time vencedor;
3. Goleada, que recebe um jogo e retorna 1, se houve uma goleada e 0 em caso contrário. Uma goleada é um placar com diferença maior ou igual a 3.

|  |
| --- |
|  |