**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.

|  |
| --- |
| // raceRecord.h  typedef struct RaceRecord TRaceRecord;  TRaceRecord \*neww(int quantity);  float calculationAverage(TRaceRecord \*raceRecord, int quantityForCalculation);  float biggerFrequency(TRaceRecord \*raceRecord, int vector);  void findByType(TRaceRecord \*raceRecord, int vector, int defineReturn);  // receRecord.c  #include <stdio.h>  #include <stdlib.h>  #include "raceRecord.h"  struct RaceRecord {  float frequency;  float temperature;  float elevation;  };  TRaceRecord \*neww(int quantity) {  TRaceRecord \*raceRecord = (TRaceRecord \*)malloc(quantity\* sizeof(TRaceRecord));  return raceRecord;  };  float calculationAverage(TRaceRecord \*raceRecord, int quantityForCalculation) {  int i; float sum = 0;  for(i = 0; i < quantityForCalculation; i++) {  sum = sum + raceRecord[i].frequency;  }  return sum / quantityForCalculation;  }  float biggerFrequency(TRaceRecord \*raceRecord, int vector) {  int i; float higherValue = 0;  for(i = 0; i < vector; i++) {  if(raceRecord[i].frequency > higherValue) {  higherValue = raceRecord[i].frequency;  }  }  return higherValue;  }  void findByType(TRaceRecord \*raceRecord, int vector, int defineReturn) {  int i;  switch(defineReturn) {  case 1:  printf("Lista de FREQUENCIAS");  for(i = 0; i < vector; i++) {  printf("\nFrequencia: %f", raceRecord[i].frequency);  } break;  case 2:  printf("Lista de TEMPERATURAS");  for(i = 0; i < vector; i++) {  printf("\nTemperatura: %f", raceRecord[i].temperature);  } break;  case 3:  printf("Lista de ELECACOES");  for(i = 0; i < vector; i++) {  printf("\nElevacao: %f", raceRecord[i].elevation);  } 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.

|  |
| --- |
| //Soccer.h  typedef struct Soccer TSoccer;  TSoccer\* newSoccer(char \*team1, char \*team2, int soccer1, int soccer2);  //TSoccer\* newSoccer(char team1[30], char team2[30], int soccer1, int soccer2);  TSoccer\* winner(TSoccer \*soccer);  int rout(TSoccer \*soccer);  // Soccer.c  #include <stdio.h>  #include <stdlib.h>  #include "Soccer.h"  typedef struct {  int soccer1, soccer2;  }TScoreboard;  struct Soccer {  char \*team1;  char \*team2;  TScoreboard scoreboard;  };  TSoccer\* newSoccer(char \*team1, char \*team2, int soccer1, int soccer2) {  TSoccer \*s = (TSoccer \*)malloc(sizeof(TSoccer));  s -> team1 = team1;  s -> team2 = team2;  s -> scoreboard.soccer1 = soccer1;  s -> scoreboard.soccer2 = soccer2;  }  TSoccer\* winner(TSoccer \*soccer) {  if(soccer -> scoreboard.soccer1 > soccer -> scoreboard.soccer2) {  printf("Vencedor: %s\n", soccer -> scoreboard.soccer1);  } else if(soccer -> scoreboard.soccer2 > soccer -> scoreboard.soccer1) {  printf("Vencedor: %s\n", soccer -> scoreboard.soccer2);  } else  printf("Empate...!\n");  }  int rout(TSoccer \*soccer) {  if(soccer -> scoreboard.soccer1 - soccer -> scoreboard.soccer2 >= 3 ||  soccer -> scoreboard.soccer2 - soccer -> scoreboard.soccer1 >= 3) {  return 1;  } else {  return 0;  }  } |