**Atividades do conteúdo: Abertura**

**0 - 1** Ao final dos seus treinos, um maratonista possui algumas informações registradas em seu smartwatch. Basicamente, a cada 5min, o relógio registra a sua frequência cardíaca, sua temperatura corporal e a elevação do terreno. Ele gostaria de um programa onde ele pudesse digitar esses os dados para avaliação, para um treino de 1h. Desenvolva um programa C com os seguintes requisitos.

**ENTRADA**:

- Leia do usuário os 12 valores de frequência cardíaca, temperatura corporal e a elevação do terreno, registrados ao longo de 1h de treino.

**SAÍDA**:

- Listagem das frequências cardíacas;

- Listagem das temperaturas corporais;

- Listagem das elevações de terreno;

- A média da sua temperatura corporal;

- A maior frequência cardíaca registrada;

- A diferença de elevação de terreno final e inicial.

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  /\*Programa para saber a media de temperatura corporal, a maior frequencia cardiaca  registrada e a diferenca de elevação de terreno inicial e final.\*/  int main(int argc, char \*argv[]) {  const int qtd = 3;  float heartRate[qtd], bodyTemperature[qtd], terrainElevation [qtd], average;  float sumTemperature, biggerFrequency, difference;  int i;    printf("\t\t-- Informe os dados a seguir -- \n");  printf("\n");  for(i=0; i<qtd; i++){  printf("Informe sua frequencia cardiaca: \n", i);  scanf("%f",&heartRate[i]);  printf("Informe sua temperatura: \n", i);  scanf("%f",&bodyTemperature[i]);  printf("Informe a elevacao do terreno: \n", i);  scanf("%f",&terrainElevation[i]);    bodyTemperature[i] = bodyTemperature[i] + sumTemperature;    if (heartRate[i] > biggerFrequency){  biggerFrequency = heartRate[i];  }  }    difference = terrainElevation [0] - terrainElevation[qtd-1];  sumTemperature = sumTemperature / qtd;  printf("\n");  printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  printf("\n");  printf("A media da temperatura e: %f\n", sumTemperature);  printf("Sua maior frequencia cardiaca e: %f\n", biggerFrequency);  printf("A diferenca de elevacao final referente a inicial e: %f\n", biggerFrequency);  return 0;  } |

**0 - 2** Inspirado(a) na primeira atividade, desenvolva um programa C com os seguintes requisitos.

**TIPOS DE DADOS (STRUCT):**

- *RegistroCorrida*, comfrequência cardíaca, temperatura corporal e a elevação do terreno;

**FUNÇÕES**:

- *ListarRegistroCorrida*, que recebe um vetor de RegistroCorrida, a quantidade de registros e o número de informação a ser exibida (1, 2 ou 3). A função mostra na tela a lista de frequências cardíacas do vetor se este último parâmetro for 1, a lista de temperaturas corporais do vetor se o parâmetro for 2 e a lista de elevações de terreno do vetor se o parâmetro for 3.

- *CalculaMedia*, que recebe um vetor de números e a sua quantidade, e retorna a média dos seus valores.

- *CalculaMaior*, que recebe um vetor de números e a sua quantidade, e retorna o seu maior valor.

**ENTRADA**:

- Leia do usuário os 12 valores de frequência cardíaca, temperatura corporal e a elevação do terreno;

**SAÍDA**:

- Listagem das frequências cardíacas;

- Listagem das temperaturas corporais;

- Listagem das elevações de terreno;

- A média da sua temperatura corporal;

- A maior frequência cardíaca registrada;

- A diferença de elevação de terreno final e inicial.

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  /\* Programa para calcular a Frequencia a temperatura e a elevelção com struct e function. \*/  const int qtd = 3;  struct raceRegistration {  float frequency, temperature, elevation;  };  void listRegistrationRace(struct raceRegistration \*rr, int n, int info) {  int i;  switch (info){  case 1: for(i=0; i<n; i++){  printf("%f\n", rr[i].frequency);  } break;  case 2: for(i=0; i<n; i++){  printf("%f\n", rr[i].temperature);  } break;  case 3: for(i=0; i<n; i++){  printf("%f\n", rr[i].elevation);  } break;  default: printf("Valor Invalido");  }  }  float calcAverage(struct raceRegistration \*rr, int n){  int i; float sum = 0;  for(i=0; i<n; i++){  sum = sum + rr[i].frequency;  }  return sum / n;  }  float calcBigger(struct raceRegistration \*rr, int n){  int i; float bigger = 0;  for(i=0; i<n; i++){  if(rr[i].temperature > bigger){  bigger = rr[i].temperature;  }  }  return bigger;  }  int main(int argc, char \*argv[]) {  int i;  struct raceRegistration \*registration;  registration = malloc(qtd\* sizeof(struct raceRegistration));  printf("==================================================================\n");  printf("\t\t-- Digite os dados -- \n");  for(i=0; i<qtd; i++){  printf("Frequencia: \n");  scanf("%f",&registration[i].frequency);  printf("Temperatura: \n");  scanf("%f",&registration[i].temperature);  printf("Elevacao: \n");  scanf("%f",&registration[i].elevation);  }  listRegistrationRace(registration, qtd, 1);  listRegistrationRace(registration, qtd, 2);  listRegistrationRace(registration, qtd, 3);  printf("==================================================================\n");  printf("Media frequencia: %f\n",calcAverage(registration, qtd));  printf("==================================================================\n");  printf("Media temperatura: %f\n",calcBigger(registration, qtd));  printf("==================================================================");  return 0;  } |

**0 - 3** Desenvolva um programa C que, baseado no código a seguir, realize as seguintes tarefas:

int a, \*b;

1. Leia do usuário um valor a ser atribuído à variável **a**;
2. Faça o ponteiro **b** apontar para **a**;
3. Dobre o valor de **a**, utilizando o ponteiro **b**.
4. Aloque nova memória para **b**;
5. Leia do usuário um valor a ser atribuído à posição de memória apontada por **b**.
6. Mostre os valores de **a** e **b** na tela.

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  /\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/  int main(int argc, char \*argv[]) {  int a, \*b;  printf("Informe o valor: ");  scanf("%i",&a);  b =&a;  \*b = 2\*(\*b);  b = malloc(sizeof(a));  scanf("%i",b);    printf("A: \n %i", a);  printf("B: \n %i", \*b);    return 0;  } |