**SFWRENG 2MP3 – Programming for Mechatronics Fall 2018**

|  |  |
| --- | --- |
| **Exercise 6 Solution** | **Submitted By: HARNEET SINGH, #400110275, singhh76** |
| **Question #** | **Answer** |
| **Entire Code for Question #1** | #include <stdio.h>  #define length 30  struct student\_record  {  char first\_name[length];  char last\_name[length];  int student\_number;  char field[length];  char program[length];  float twelve\_PointGPA;  };  float four\_PointGPA(float GPA12);  void printStudentInfo();  void main()  {  struct student\_record FirstEntry;  struct student\_record \*ptrStudent = &FirstEntry;  puts("This program stores student records and computes 4-point GPA:");  printf("Please enter the first name of the student: ");  scanf("%s", ptrStudent->first\_name);  printf("Please enter the last name of the student: ");  scanf("%s", ptrStudent->last\_name);  printf("Please enter the student number: ");  scanf("%d", &(ptrStudent->student\_number)); //second method shown later in the code  printf("Please enter the field of the student (e.g. engineering/management): ");  scanf("%s", ptrStudent->field);  printf("Please enter the program of the student (e.g. software): ");  scanf("%s", ptrStudent->program);  restart:  printf("Please enter the 12-point GPA of the student: ");  scanf("%f", &(\*ptrStudent).twelve\_PointGPA);  float convertedGPA = -1; //done to avoid any garbage value  convertedGPA = four\_PointGPA((\*ptrStudent).twelve\_PointGPA);  if (convertedGPA != -1)  {  printf("\n\*\*\*\* The 4-point GPA of the student is: %.1f \*\*\*\*\n", convertedGPA);  }  else  {  printf("!!!Please enter a valid 12-point GPA\n");  goto restart;  }  printStudentInfo(ptrStudent); //passing struct variable's address  }  float four\_PointGPA(float GPA12)  {  if (10.9 <= GPA12 && GPA12<= 12.0) {  return 4.0; }  else if (10.6 <= GPA12 && GPA12<= 10.8) {  return 3.9; }  else if (10.3 <= GPA12 && GPA12 <= 10.5) {  return 3.8; }  else if (10.0 <= GPA12 && GPA12 <= 10.2) {  return 3.7; }  else if (9.7 <= GPA12 && GPA12 <= 9.9) {  return 3.6; }  else if (9.4 <= GPA12 && GPA12 <= 9.6) {  return 3.5; }  else if (9.1 <= GPA12 && GPA12 <= 9.3) {  return 3.4; }  else if (8.8 <= GPA12 && GPA12 <= 9.0) {  return 3.3; }  else if (8.5 <= GPA12 && GPA12 <= 8.7) {  return 3.2; }  else if (8.2 <= GPA12 && GPA12<= 8.4) {  return 3.1; }  else if (7.9 <= GPA12 && GPA12 <= 8.1) {  return 3.0; }  else if (7.6 <= GPA12 && GPA12<= 7.8) {  return 2.9; }  else if (7.3 <= GPA12 && GPA12 <= 7.5) {  return 2.8; }  else if (7.0 <= GPA12 && GPA12 <= 7.2) {  return 2.7; }  else if (6.7 <= GPA12 && GPA12 <= 6.9) {  return 2.6; }  else if (6.4 <= GPA12 && GPA12 <= 6.6) {  return 2.5; }  else if (6.1 <= GPA12 && GPA12 <= 6.3) {  return 2.4; }  else if (5.8 <= GPA12 && GPA12 <= 6.0) {  return 2.3; }  else if (5.5 <= GPA12 && GPA12 <= 5.7) {  return 2.2; }  else if (5.2 <= GPA12 && GPA12 <= 5.4) {  return 2.1; }  else if (4.9 <= GPA12 && GPA12 <= 5.1) {  return 2.0; }  else if (4.6 <= GPA12 && GPA12 <= 4.8) {  return 1.9; }  else if (4.3 <= GPA12 && GPA12 <= 4.5) {  return 1.8; }  else if (4.0 <= GPA12 && GPA12 <= 4.2) {  return 1.7; }  else if (3.7 <= GPA12 && GPA12 <= 3.9) {  return 1.6; }  else if (3.4 <= GPA12 && GPA12 <= 3.6) {  return 1.5; }  else if (3.1 <= GPA12 && GPA12 <= 3.3) {  return 1.4; }  else if (2.8 <= GPA12 && GPA12 <= 3.0) {  return 1.3; }  else if (2.5 <= GPA12 && GPA12 <= 2.7) {  return 1.2; }  else if (2.2 <= GPA12 && GPA12 <= 2.4) {  return 1.1; }  else if (1.8 <= GPA12 && GPA12 <= 2.1) {  return 1.0; }  else if (1.5 <= GPA12 && GPA12 <= 1.7) {  return 0.9; }  else if (1.3 <= GPA12 && GPA12 <= 1.5) {  return 0.8; }  else if (1.0 <= GPA12 && GPA12 <= 1.2) {  return 0.7; }  else if (GPA12 == 0.9) {  return 0.6; }  else if (0.7 <= GPA12 && GPA12 <= 0.8) {  return 0.5; }  else if (GPA12 == 0.6) {  return 0.4; }  else if (0.4 <= GPA12 && GPA12 <= 0.5) {  return 0.3; }  else if (GPA12 == 0.3) {  return 0.2; }  else if (0.1 <= GPA12 && GPA12 <= 0.2) {  return 0.1;}  else if (GPA12 == 0.0) {  return 0.0; }  else {return -1.0; }  }  void printStudentInfo(struct student\_record \*ptrToInfo)  {  puts("\nStudent's record shows the following details:\n");  printf("Student's first name: \t\t%s\n", (\*ptrToInfo).first\_name);  printf("Student's last name: \t\t%s\n", ptrToInfo->last\_name);  printf("Student's student number: \t%d\n", (\*ptrToInfo).student\_number);  printf("Student's program name: \t%s %s\n", (\*ptrToInfo).program, (\*ptrToInfo).field);  printf("Student's 12-point GPA: \t%.1f\n", (\*ptrToInfo).twelve\_PointGPA);  } |
|  |  |
| **#1-a** | #include <stdio.h>  #define length 30  struct student\_record  {  char first\_name[length];  char last\_name[length];  int student\_number;  char field[length];  char program[length];  float twelve\_PointGPA;  };  float four\_PointGPA(float GPA12);  void printStudentInfo(); |
|  |
| **#1-b** | void main()  {  struct student\_record FirstEntry;  struct student\_record \*ptrStudent = &FirstEntry;  puts("This program stores student records and computes 4-point GPA:");  printf("Please enter the first name of the student: ");  scanf("%s", ptrStudent->first\_name);  printf("Please enter the last name of the student: ");  scanf("%s", ptrStudent->last\_name);  printf("Please enter the student number: ");  scanf("%d", &(ptrStudent->student\_number)); //second method to do the same-shown in line 31  printf("Please enter the field of the student (e.g. engineering/management): ");  scanf("%s", ptrStudent->field);  printf("Please enter the program of the student (e.g. software): ");  scanf("%s", ptrStudent->program);  restart:  printf("Please enter the 12-point GPA of the student: ");  scanf("%f", &(\*ptrStudent).twelve\_PointGPA);  float convertedGPA = -1; //done to avoid any garbage value  convertedGPA = four\_PointGPA((\*ptrStudent).twelve\_PointGPA);  if (convertedGPA != -1)  {  printf("\n\*\*\*\* The 4-point GPA of the student is: %.1f \*\*\*\*\n", convertedGPA);  }  else  {  printf("!!!Please enter a valid 12-point GPA\n");  goto restart;  }  printStudentInfo(ptrStudent); //passing struct variable's address  } |
|  |
| **#1-c** | float four\_PointGPA(float GPA12)  {  if (10.9 <= GPA12 && GPA12<= 12.0) {  return 4.0; }  else if (10.6 <= GPA12 && GPA12<= 10.8) {  return 3.9; }  else if (10.3 <= GPA12 && GPA12 <= 10.5) {  return 3.8; }  else if (10.0 <= GPA12 && GPA12 <= 10.2) {  return 3.7; }  else if (9.7 <= GPA12 && GPA12 <= 9.9) {  return 3.6; }  else if (9.4 <= GPA12 && GPA12 <= 9.6) {  return 3.5; }  else if (9.1 <= GPA12 && GPA12 <= 9.3) {  return 3.4; }  else if (8.8 <= GPA12 && GPA12 <= 9.0) {  return 3.3; }  else if (8.5 <= GPA12 && GPA12 <= 8.7) {  return 3.2; }  else if (8.2 <= GPA12 && GPA12<= 8.4) {  return 3.1; }  else if (7.9 <= GPA12 && GPA12 <= 8.1) {  return 3.0; }  else if (7.6 <= GPA12 && GPA12<= 7.8) {  return 2.9; }  else if (7.3 <= GPA12 && GPA12 <= 7.5) {  return 2.8; }  else if (7.0 <= GPA12 && GPA12 <= 7.2) {  return 2.7; }  else if (6.7 <= GPA12 && GPA12 <= 6.9) {  return 2.6; }  else if (6.4 <= GPA12 && GPA12 <= 6.6) {  return 2.5; }  else if (6.1 <= GPA12 && GPA12 <= 6.3) {  return 2.4; }  else if (5.8 <= GPA12 && GPA12 <= 6.0) {  return 2.3; }  else if (5.5 <= GPA12 && GPA12 <= 5.7) {  return 2.2; }  else if (5.2 <= GPA12 && GPA12 <= 5.4) {  return 2.1; }  else if (4.9 <= GPA12 && GPA12 <= 5.1) {  return 2.0; }  else if (4.6 <= GPA12 && GPA12 <= 4.8) {  return 1.9; }  else if (4.3 <= GPA12 && GPA12 <= 4.5) {  return 1.8; }  else if (4.0 <= GPA12 && GPA12 <= 4.2) {  return 1.7; }  else if (3.7 <= GPA12 && GPA12 <= 3.9) {  return 1.6; }  else if (3.4 <= GPA12 && GPA12 <= 3.6) {  return 1.5; }  else if (3.1 <= GPA12 && GPA12 <= 3.3) {  return 1.4; }  else if (2.8 <= GPA12 && GPA12 <= 3.0) {  return 1.3; }  else if (2.5 <= GPA12 && GPA12 <= 2.7) {  return 1.2; }  else if (2.2 <= GPA12 && GPA12 <= 2.4) {  return 1.1; }  else if (1.8 <= GPA12 && GPA12 <= 2.1) {  return 1.0; }  else if (1.5 <= GPA12 && GPA12 <= 1.7) {  return 0.9; }  else if (1.3 <= GPA12 && GPA12 <= 1.5) {  return 0.8; }  else if (1.0 <= GPA12 && GPA12 <= 1.2) {  return 0.7; }  else if (GPA12 == 0.9) {  return 0.6; }  else if (0.7 <= GPA12 && GPA12 <= 0.8) {  return 0.5; }  else if (GPA12 == 0.6) {  return 0.4; }  else if (0.4 <= GPA12 && GPA12 <= 0.5) {  return 0.3; }  else if (GPA12 == 0.3) {  return 0.2; }  else if (0.1 <= GPA12 && GPA12 <= 0.2) {  return 0.1;}  else if (GPA12 == 0.0) {  return 0.0; }  else {return -1.0; }  } |
|  |
| **#1-d** | void printStudentInfo(struct student\_record \*ptrToInfo)  {  puts("\nStudent's record shows the following details:\n");  printf("Student's first name: \t\t%s\n", (\*ptrToInfo).first\_name);  printf("Student's last name: \t\t%s\n", ptrToInfo->last\_name);  printf("Student's student number: \t%d\n", (\*ptrToInfo).student\_number);  printf("Student's program name: \t%s %s\n", (\*ptrToInfo).program, (\*ptrToInfo).field);  printf("Student's 12-point GPA: \t%.1f\n", (\*ptrToInfo).twelve\_PointGPA);  } |
|  |
| **Entire Code for Question 2** | #include <stdio.h>  #define length 30  struct SFWRENG\_MP3\_Students {  char firstName[length];  char lastName[length];  int studentNumber;  float grade;  };  void MeanMedianMode();  void printRecord();  void main ()  {  int totalStudents = 0;  puts("\nThis program creates record of students enrolled in SFWRENG-MP3:");  printf("Please enter the number of students needed in the record: ");  scanf("%d", &totalStudents);  struct SFWRENG\_MP3\_Students student[totalStudents];  struct SFWRENG\_MP3\_Students \*ptrStudent = student;  for (int j = 0; j < totalStudents; j++, ptrStudent++) //storing values  {  printf("Please enter first name of the student #%d:", j+1);  scanf("%s", ptrStudent->firstName);  printf("Please enter last name of the student #%d:", j+1);  scanf("%s", ptrStudent->lastName);  printf("Please enter student number of the student #%d:", j+1);  scanf("%d", &ptrStudent->studentNumber);  printf("Please enter course grade of the student #%d:", j+1);  scanf("%f", &ptrStudent->grade);  }  ptrStudent = ptrStudent - totalStudents;  MeanMedianMode(ptrStudent, totalStudents);  printRecord(ptrStudent, totalStudents);  }  void MeanMedianMode(struct SFWRENG\_MP3\_Students \*ptrgrade, int numberOfStudents)  {  float sum = 0;  float score[numberOfStudents]; //array to store grades of all the students  for (int i = 0; i < numberOfStudents; i++, ptrgrade++)  {  sum = sum + (\*ptrgrade).grade;  score[i] = ptrgrade->grade;  }  ptrgrade = ptrgrade - numberOfStudents;  float mean = (sum / numberOfStudents);  printf("\n\t\*\*\*\*\*\*\*\* \nMean for this course: %.2f\n", mean);  for (int j = 1; j < numberOfStudents; j++) //sorting array  {  for (int k = 0; k < numberOfStudents - 1; k++)  {  if (score[k] > score[k+1])  {  float temp = score[k];  score[k] = score[k+1];  score[k+1] = temp;  }  }  }  if (numberOfStudents%2 == 0)  {  int middle = numberOfStudents/2;  float median = ((score[middle-1])+(score[middle]))/2;  printf("Median for this course: %.2f\n", median);  }  else  {  printf("Median for this course: %.2f\n", score[(numberOfStudents-1)/2]);  }  int frequency = 0, a, b;  float mode;  for(int s = 0; s < numberOfStudents; s++)  {  float firstValue = score[s];  int count = 1;  for (int p = s + 1; p < numberOfStudents; p++)  {  a = (int)((firstValue\*100)/100);  b = (int)(((score[p])\*100)/100);  if (a == b)  {  count++;  }  }  if(count > frequency)  {  frequency = count;  mode = firstValue;  }  }  printf("Mode for this course: %.2f with frequency of %d\n\t\*\*\*\*\*\*\*\*\n",  mode, frequency);  }  void printRecord(struct SFWRENG\_MP3\_Students \*ptrgrade, int numberOfStudents)  {  for (int j = 0; j < numberOfStudents; j++, ptrgrade++)  {  printf("Name of Student #%d: \t\t%s %s\n", j+1, ptrgrade->firstName,  ptrgrade->lastName);  printf("Student number of student #%d: \t%d\n", j+1, ptrgrade->studentNumber);  printf("Course grade of student #%d: \t%.2f\n\n", j+1, ptrgrade->grade);  }  } |
|  |  |
| **#2-a** | #include <stdio.h>  #define length 30  struct SFWRENG\_MP3\_Students {  char firstName[length];  char lastName[length];  int studentNumber;  float grade;  };  void MeanMedianMode();  void printRecord(); |
|  |
| **#2-b** | void main ()  {  int totalStudents = 0;  puts("\nThis program creates record of students enrolled in SFWRENG-MP3:");  printf("Please enter the number of students needed in the record: ");  scanf("%d", &totalStudents);  struct SFWRENG\_MP3\_Students student[totalStudents];  struct SFWRENG\_MP3\_Students \*ptrStudent = student;  for (int j = 0; j < totalStudents; j++, ptrStudent++) //storing values  {  printf("Please enter first name of the student #%d:", j+1);  scanf("%s", ptrStudent->firstName);  printf("Please enter last name of the student #%d:", j+1);  scanf("%s", ptrStudent->lastName);  printf("Please enter student number of the student #%d:", j+1);  scanf("%d", &ptrStudent->studentNumber);  printf("Please enter course grade of the student #%d:", j+1);  scanf("%f", &ptrStudent->grade);  }  ptrStudent = ptrStudent - totalStudents;  MeanMedianMode(ptrStudent, totalStudents);  printRecord(ptrStudent, totalStudents);  } |
|  |
| **#2-c** | void MeanMedianMode(struct SFWRENG\_MP3\_Students \*ptrgrade, int numberOfStudents)  {  float sum = 0;  float score[numberOfStudents]; //array to store grades of all the students  for (int i = 0; i < numberOfStudents; i++, ptrgrade++)  {  sum = sum + (\*ptrgrade).grade;  score[i] = ptrgrade->grade;  }  ptrgrade = ptrgrade - numberOfStudents;  float mean = (sum / numberOfStudents);  printf("\n\t\*\*\*\*\*\*\*\* \nMean for this course: %.2f\n", mean);  for (int j = 1; j < numberOfStudents; j++) //sorting array  {  for (int k = 0; k < numberOfStudents - 1; k++)  {  if (score[k] > score[k+1])  {  float temp = score[k];  score[k] = score[k+1];  score[k+1] = temp;  }  }  }  if (numberOfStudents%2 == 0)  {  int middle = numberOfStudents/2;  float median = ((score[middle-1])+(score[middle]))/2;  printf("Median for this course: %.2f\n", median);  }  else  {  printf("Median for this course: %.2f\n", score[(numberOfStudents-1)/2]);  }  int frequency = 0, a, b;  float mode;  for(int s = 0; s < numberOfStudents; s++)  {  float firstValue = score[s];  int count = 1;  for (int p = s + 1; p < numberOfStudents; p++)  {  a = (int)((firstValue\*100)/100);  b = (int)(((score[p])\*100)/100);  if (a == b)  {  count++;  }  }  if(count > frequency)  {  frequency = count;  mode = firstValue;  }  }  printf("Mode for this course: %.2f with frequency of %d\n\t\*\*\*\*\*\*\*\*\n",  mode, frequency);  } |
|  |
| **#2-d** | void printRecord(struct SFWRENG\_MP3\_Students \*ptrgrade, int numberOfStudents)  {  for (int j = 0; j < numberOfStudents; j++, ptrgrade++)  {  printf("Name of Student #%d: \t\t%s %s\n", j+1, ptrgrade->firstName,  ptrgrade->lastName);  printf("Student number of student #%d: \t%d\n", j+1, ptrgrade->studentNumber);  printf("Course grade of student #%d: \t%.2f\n\n", j+1, ptrgrade->grade);  }  } |
|  |