# HOA

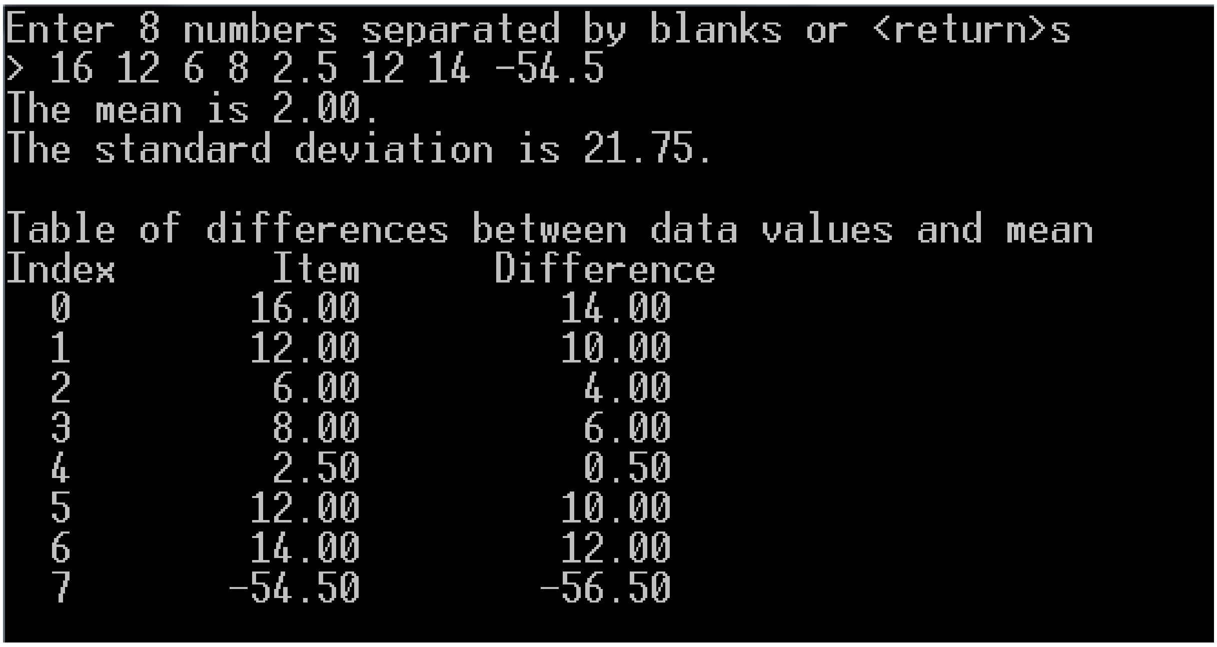
## QUESTION 1

Arrays are commonly used to store a collection of relation data values. Once the values are stored, you can perform simple statistical computations. Given the below equations, write a program that prints a table of differences (see sample output).

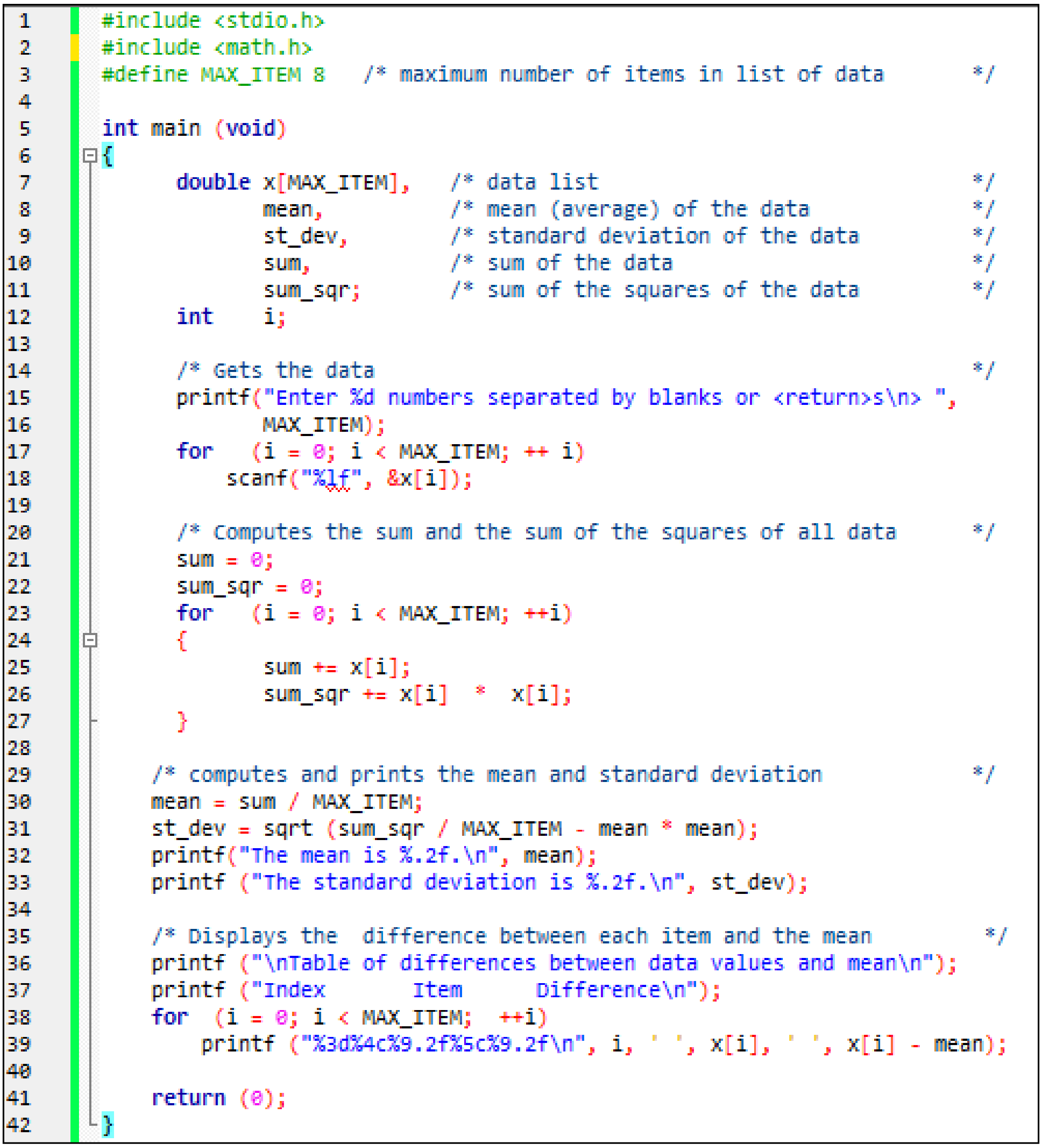
$$sum = \mathtt{x[0]} + \mathtt{x[1]} + \cdots + \mathtt{x[6]} + \mathtt{x[7]} = \sum_{\mathtt{i=0}}^{\mathtt{MAX\_ITEM - 1}} \mathtt{x[i]}$$

$$sum\_sqr = \mathtt{x[0]}^2 + \mathtt{x[1]}^2 + \cdots + \mathtt{x[6]}^2 + \mathtt{x[7]}^2 = \sum_{\mathtt{i=0}}^{\mathtt{MAX\_ITEM - 1}} \mathtt{x[i]}^2$$

$$standard\ deviation = \sqrt{\frac{\sum_{\mathtt{i=0}}^{\mathtt{MAX\_ITEM-1}} \mathtt{x[i]}^2}{\mathtt{MAX\_ITEM}} - mean^2}$$



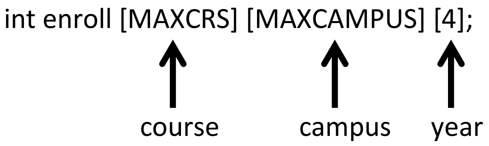
```
Enter 8 numbers separated by blanks or <return>s
> 16 12 6 8 2.5 12 14 -54.5
The mean is 2.00.
The standard deviation is 21.75.

Table of differences between data values and mean
Index       Item      Difference
  0        16.00         14.00
  1        12.00         10.00
  2         6.00          4.00
  3         8.00          6.00
  4         2.50          0.50
  5        12.00         10.00
  6        14.00         12.00
  7       -54.50        -56.50
```



```c
#include <stdio.h>
#include <math.h>
#define MAX_ITEM 8   /* maximum number of items in list of data       */

int main (void)
{
      double x[MAX_ITEM],   /* data list                              */
             mean,          /* mean (average) of the data             */
             st_dev,        /* standard deviation of the data         */
             sum,           /* sum of the data                        */
             sum_sqr;       /* sum of the squares of the data         */
      int    i;

      /* Gets the data                                                */
      printf("Enter %d numbers separated by blanks or <return>s\n> ",
             MAX_ITEM);
      for   (i = 0; i < MAX_ITEM; ++ i)
          scanf("%lf", &x[i]);

      /* Computes the sum and the sum of the squares of all data      */
      sum = 0;
      sum_sqr = 0;
      for   (i = 0; i < MAX_ITEM; ++i)
      {
             sum += x[i];
             sum_sqr += x[i]  *  x[i];
      }

    /* computes and prints the mean and standard deviation            */
    mean = sum / MAX_ITEM;
    st_dev = sqrt (sum_sqr / MAX_ITEM - mean * mean);
    printf("The mean is %.2f.\n", mean);
    printf ("The standard deviation is %.2f.\n", st_dev);

    /* Displays the  difference between each item and the mean         */
    printf ("\nTable of differences between data values and mean\n");
    printf ("Index       Item      Difference\n");
    for  (i = 0; i < MAX_ITEM;  ++i)
        printf ("%3d%4c%9.2f%5c%9.2f\n", i, ' ', x[i], ' ', x[i] - mean);

    return (0);
}
```

## QUESTION 2

Assume that you have a three dimensional array ***enroll*** which keeps the number of offered courses, number of campuses and number of class years (freshman:0, sophomore:1, junior:2, senior:3) in a university as below:

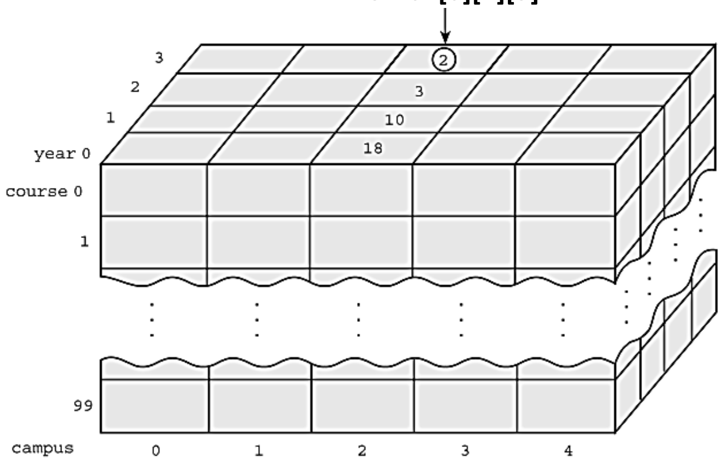


int enroll [MAXCRS] [MAXCAMPUS] [4];

course campus year

For instance, enroll[0][2][3] gives you the number of seniors taking course 0 at campus 2 (see below image).

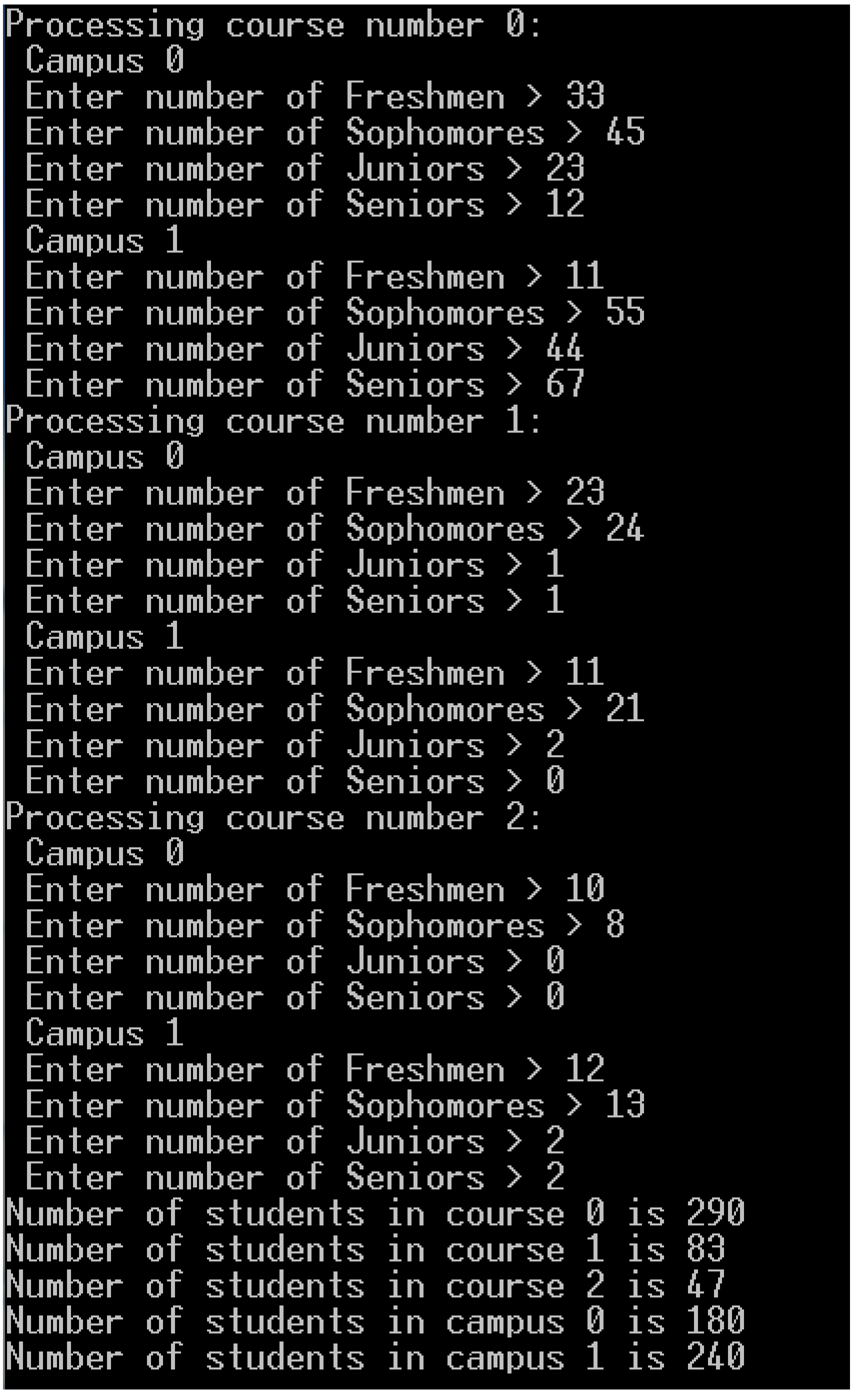
**enroll[0][2][3]**



Given the following main function and sample output, write the missing functions.

```
#include <stdio.h>
#define MAXCRS 3
#define MAXCAMPUS 2

void enrollStudents(int arr[MAXCRS][MAXCAMPUS][4]);
void displayNoStudentsInEachCourse(const int arr[MAXCRS][MAXCAMPUS][4]);
void displayNoStudentsInEachCampus(const int arr[MAXCRS][MAXCAMPUS][4]);
int find_students(int arr[MAXCRS][MAXCAMPUS][4], int rank, int course);

int main(void)
{
    int enroll [MAXCRS] [MAXCAMPUS] [4];

    enrollStudents(enroll);
    displayNoStudentsInEachCourse(enroll);
    displayNoStudentsInEachCampus(enroll);

    return 0;
}
```

*find_students function finds the number of students of the given rank who are enrolled in the given course on all campuses.



```
Processing course number 0:
 Campus 0
 Enter number of Freshmen > 33
 Enter number of Sophomores > 45
 Enter number of Juniors > 23
 Enter number of Seniors > 12
 Campus 1
 Enter number of Freshmen > 11
 Enter number of Sophomores > 55
 Enter number of Juniors > 44
 Enter number of Seniors > 67
Processing course number 1:
 Campus 0
 Enter number of Freshmen > 23
 Enter number of Sophomores > 24
 Enter number of Juniors > 1
 Enter number of Seniors > 1
 Campus 1
 Enter number of Freshmen > 11
 Enter number of Sophomores > 21
 Enter number of Juniors > 2
 Enter number of Seniors > 0
Processing course number 2:
 Campus 0
 Enter number of Freshmen > 10
 Enter number of Sophomores > 8
 Enter number of Juniors > 0
 Enter number of Seniors > 0
 Campus 1
 Enter number of Freshmen > 12
 Enter number of Sophomores > 13
 Enter number of Juniors > 2
 Enter number of Seniors > 2
Number of students in course 0 is 290
Number of students in course 1 is 83
Number of students in course 2 is 47
Number of students in campus 0 is 180
Number of students in campus 1 is 240
```

```c
/*
     * Compute the number of students in a course who have a
     * specific rank.
     * returns -1 if rank or course is out of range.
     */
int find_students (int arr[MAXCRS][MAXCAMPUS][4], int rank, int course)
{
    int i, cnt = 0;

    if ((rank >= 0 && rank <= 3) && (course >= 0 && course < MAXCRS))
        for (i = 0; i < MAXCAMPUS; ++i)
            cnt += arr[course][i][rank];
    else
           cnt = -1;

    return (cnt);
}

void enrollStudents(int arr[MAXCRS][MAXCAMPUS][4])
{
    int i, j, k;
    for (i = 0; i < MAXCRS; ++i)
    {
        printf("Processing course number %d: \n", i);
        for (j = 0; j < MAXCAMPUS; ++j)
        {
            printf(" Campus %d\n", j);

            for (k = 0; k < 4; ++k)
            {
                printf(" Enter number of ");
                switch (k)
                {
                  case 0 :
                          printf("Freshmen > ");
                          break;
                  case 1 :
                          printf("Sophomores > ");
                          break;
                  case 2 :
                          printf("Juniors > ");
                          break;
                  case 3 :
                          printf("Seniors > ");
                }
                  scanf("%d", &arr[i][j][k]);
            }
        }
    }

}
```

```
void displayNoStudentsInEachCourse(const int arr[MAXCRS][MAXCAMPUS][4])
{
    int crs_sum;
    for(int course = 0; course < MAXCRS; ++ course)
    {
        crs_sum = 0;
        for(int campus = 0; campus < MAXCAMPUS; ++campus)
        {
            for(int class_rank = 0; class_rank < 4; ++class_rank)
            {
                crs_sum += arr[course][campus][class_rank];
            }
        }
        printf("Number of students in course %d is %d\n", course, crs_sum);
    }
}

void displayNoStudentsInEachCampus(const int arr[MAXCRS][MAXCAMPUS][4])
{
    int camp_sum;
    for(int campus = 0; campus < MAXCAMPUS; ++campus)
    {
        camp_sum = 0;
        for(int course = 0; course < MAXCRS; ++ course)
        {
            for(int class_rank = 0; class_rank < 4; ++class_rank)
            {
                camp_sum += arr[course][campus][class_rank];
            }
        }
        printf("Number of students in campus %d is %d\n", campus,
camp_sum);
    }
}
```