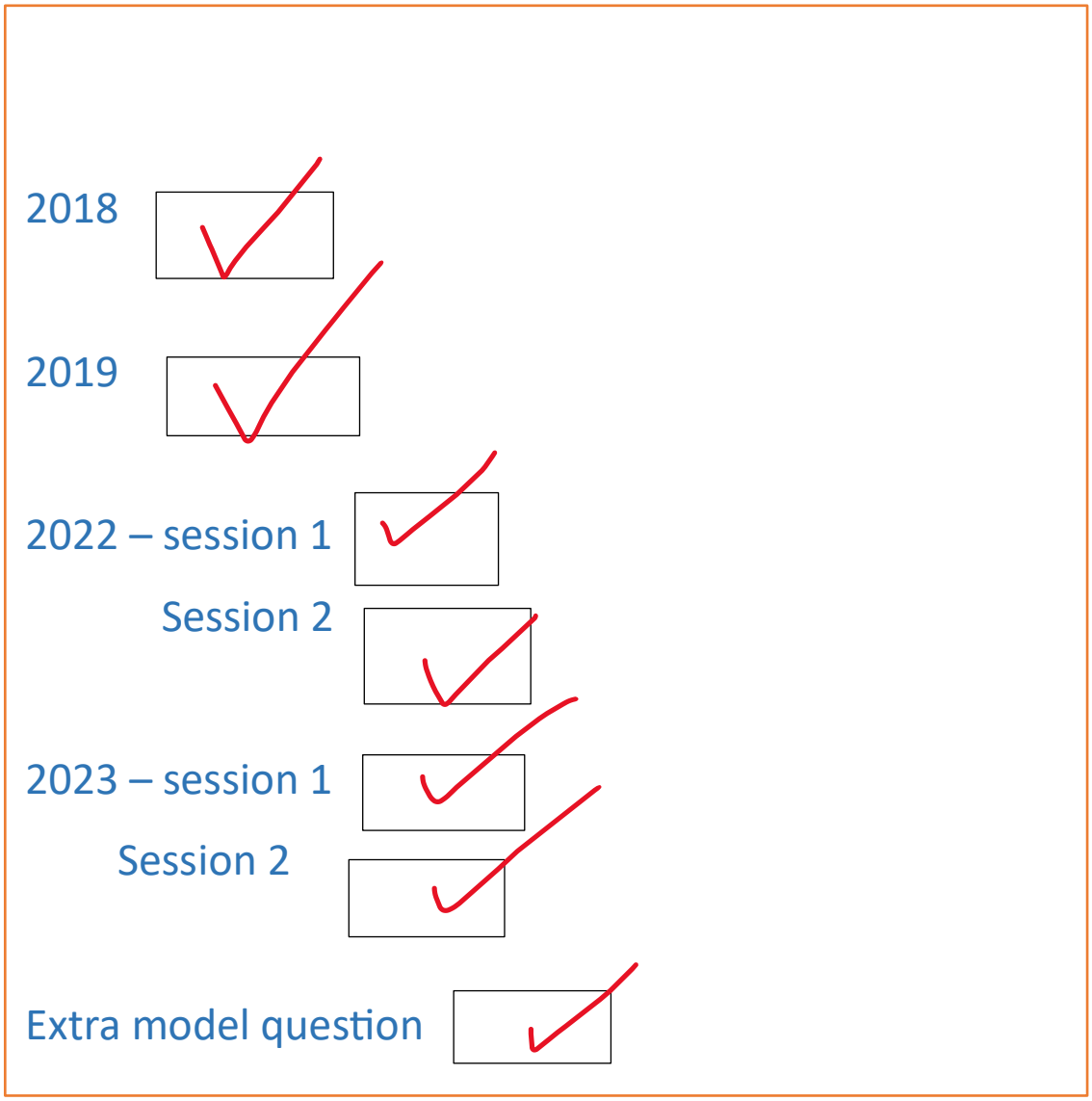
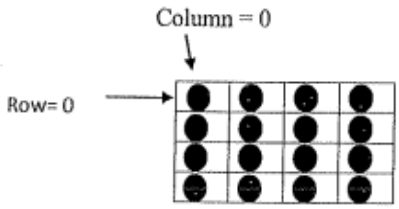
Arrays Past paper Questions



2018

PART B

A 4 x 4 square display panel consists of LED bulbs of red, green and blue colours. Write a C program to create a character array to represent the LED panel as shown below.



Enter the color of the LED bulbs ('R', 'G' or 'B') of the panel from the keyboard. Display the colours of the LED display.
Go through the array and display the position (row and column) of the "Red" bulbs.

Sample output is given below

R R G B
G G G B
R G B B
B B B G

Red LED bulb positions
[0,0] [0,1] [2,0]

Save your program as Save your program as 1AJune2b.c

```c
#include <stdio.h>

int main(){

    int i, j, k, redCount = 0;
    char colour[4][4];
    int positions[4][2];

    for(i=0; i<4; i++){

        for(j=0; j<4; j++){
            printf("Input colour(R,G,B) of bulb %d: ", j+1);
            scanf(" %c",&colour[i][j]);
        }
        printf("\n");

    }
    for(i=0; i<4; i++){
        for (j=0; j<4; j++){
            printf("%c ",colour[i][j]);
            if(colour[i][j] == 'R'){
                positions[redCount][0] = i;
                positions[redCount][1] = j;
                redCount++;
            }
        }
        printf("\n");
    }

    printf("\n\nRed LED bulb Positions\n");

    for(k=0; k < redCount ; k++){

        printf("[%d,%d] ", positions[k][0], positions[k][1] );

    }

    return 0;
}
```

2019

Question 2 **(30 marks)**

Part 1

Write a C program to fill the following array to represent a Fibonaci series.

0	1	1	2	3	5	8	13	21	34

a) Create an array called **fib** of size 10.
b) Initialize first two array elements with 0 and 1 respectively.
c) Fill the other array elements as follows using a **repetition** structure.
   fib[2] = fib[0] + fib[1]
   fib[3] = fib[1] + fib[2]
   fib[4] = fib[2] + fib[3]
   ……..
   ……..
d) Display the output as shown in the above figure.

Save your program as 1AQ2a.c

Part 2

A 2D array is used to store the details of the rating of 3 movies given by 4 reviewers. Some sample data is shown below.

		Reviewers			
		1			
	1	4	6	2	5
Movies	2	7	9	4	8
	3	6	9	3	7

Write a C program to do the following.

a) Declare an array called **ratings** to store the details of the rating of 3 movies given by 4 reviewers.
b) Read the movie ratings from the key board and store the data in the array.
c) Display the array in tabular format.
d) For each movie display the movie number and the average rating.
e) Find and display the movie which received highest average rating.

Save your program as 1AQ2b.c

Part A

```c
#include <stdio.h>

int main(){

    int fib[10] = {0,1};
    int i;

    for(i=0; i < 10; i++){
        fib[i + 2] = fib[i] + fib[i + 1];
    }

    for(i=0; i < 10 ; i++){
        printf("%d ", fib[i]);
    }

    return 0;
}
```

Part B

```c
#include <stdio.h>

int main(){

    int ratings[3][4];
    int i, j, m;
    float max = 0, average = 0;
    int total = 0;

    for(i=0; i<3; i++){
        printf("Movie %d\n", i+1);
        for(j=0; j<4; j++){
            printf("Input rating of reviewer %d: ",j+1);
            scanf("%d", &ratings[i][j]);

            total = total + ratings[i][j];
        }

        printf("\n");
    }

    for(i=0; i<3; i++){
        for(j=0; j<4; j++){
            printf("%d ", ratings[i][j]);
        }

        printf("\n");
    }

    printf("\n");

    for (i = 0; i < 3; i++) {
        total = 0; // Reset the total for each movie

        for (j = 0; j < 4; j++) {
            total += ratings[i][j];
        }
        average = (float)total / 4.0;

        if (max < average) {
            max = average;
            m = i + 1;
        }
        printf("Movie %d\t%.2f\n", i + 1, average);
    }

    printf("\nMovie %d recieved the highest average rating\n", m);

    return 0;
}
```

2022- session1

Question 2 [30 Marks]

Implement a C program to do the following.

a)

i) Create an integer array called **numArr** of size 8 and allow the user to enter values from the keyboard.

ii) Display the array elements.

iii) Move all the elements in the array **n** positions to right. **n** can be any positive number greater than zero and but less than maximum size of the array.

iv) Display the new order of array elements.

Original Array:

5	2	8	1	6	2	4	7

If n= 2, after moving all the 2 positions to right

4	7	5	2	8	1	6	2

Save your program as Q2a.c

b) Identity matrix is a square matrix in which all the elements of the principle diagonal are ones and all other elements are zero.

e.g.
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1

i) Create a two dimensional integer array called **identityArr** of size 4 x 4 and allow the user to enter values from the keyboard.

User interface should be as follows:

Values for row1
Enter element 1 :
Enter element 2 :
Enter element 3 :
Enter element 4 :
Values for row2
Enter element 1 :
Enter element 2 :
Enter element 3 :
Enter element 4 :
.....
.....

ii) Display the contents of **identityArr** array and check whether it is an identity matrix or not and display an appropriate message.

Save your program as Q2b.c

Part a

```c
#include <stdio.h>

int main(){

    int numArr[8], i, n;

    for(i=0; i<8; i++){
        printf("Number %d: ", i+1);
        scanf("%d", &numArr[i]);
    }

    printf("\n");

    for(i=0; i<8; i++){
        printf("%d ",numArr[i]);
    }

    // Input the number of positions to shift (n)
    printf("\n\nEnter the number of positions to shift to the right: ");
    scanf("%d", &n);

    // Perform the right shift
    if (n > 0 && n < 8) {
        int temp[8]; // Temporary array to store shifted values

        /*i is the current index of the element in the
         original array.
        n is the number of positions to shift to the right.
        % 8 is used to ensure that the new position wraps around
         to the beginning of the array if it exceeds the array
         size (8).
        For example, if i is 3, and n is 2, then (i + n) % 8
        would be (3 + 2) % 8, which equals 5. This means the
         element at index 3 in the original array is shifted
         to index 5 in the temporary array.*/
        for (i = 0; i < 8; i++) {
            temp[(i + n) % 8] = numArr[i];
        }
        for (i = 0; i < 8; i++) {
            numArr[i] = temp[i];
        }

        printf("\nIf n = %d , after moving all the 2 positions to the right\n", n);
        for (i = 0; i < 8; i++) {
            printf("%d ", numArr[i]);
        }
    } else {
        printf("\nInvalid input for the number of positions to shift. It should be between 1 and 7.\n");
    }

    printf("\n");

    return 0;
}
```

Part b

```c
#include <stdio.h>

int main(){

    int identityArr[4][4];
    int arr[4][4];
    int i, j;

    for(i=0; i<4; i++){
        for(j=0; j<4; j++){
            printf("Enter element %d: ",j+1);
            scanf("%d", &identityArr[i][j]);
        }
        printf("\n");
    }

    for(i=0; i<4; i++){
        for(j=0; j<4 ; j++){
            printf("%d ",identityArr[i][j] );
        }
        printf("\n");
    }

    int isIdentity = 1; // Assume it's an identity matrix

    for (i = 0; i < 4; i++) {
        for (j = 0; j < 4; j++) {
            if (i == j && identityArr[i][j] != 1) {
                isIdentity = 0; // If diagonal elements are not 1, it's not an identity matrix
            } else if (i != j && identityArr[i][j] != 0) {
                isIdentity = 0; // If non-diagonal elements are not 0, it's not an identity matrix
            }
        }
    }

    // Output the identity check result
    if (isIdentity) {
        printf("The entered matrix is a 4x4 identity matrix.\n");
    } else {
        printf("The entered matrix is not a 4x4 identity matrix.\n");
    }

    return 0;
}
```

2022- session2

Question 2 [30 Marks]

You are supposed to implement an application to count user rating about a product. Write a C program to do the following.

a)

i) Create an integer array called **userRating** of size 5.

ii) Allow the user to input any number of user ratings. User can terminate entering data when user enters -99. Rating values should be in the range of 1 to 5. Numbers not in the 1 to 5 range can be ignored.

iii) Store the count of each number in the array (e.g: how many 1s, how many 2s,….entered from the keyboard).

iv) Show the ratings as a histogram.

```
1    ************
2    **********************
3    *********
4    ******************************
5    ******************
```

Save your program as Q2a.c

00419

b) An image can be represented using a 2D array where each array element represents a gray scale value of a pixel.

i) Create a two dimensional integer array called **image** of size 4 x 4. Allow the user to enter values to the array.

User interface should be as follows:

```
Values for row1
    Enter element 1 :
    Enter element 2 :
    Enter element 3 :
    Enter element 4 :
Values for row2
    Enter element 1 :
    Enter element 2 :
    Enter element 3 :
    Enter element 4 :
.....
```

ii) Convert the gray scale image to a binary image by setting all the array elements above 55 to 1 and below 55 to 0 (zero).

iii) Display the two arrays.

e.g.

```
45  123  203  67
56  71   222  80
89  90   104  38
90  70   40   44

0  1  1  1
1  1  1  1
1  1  1  0
1  1  0  0
```

Save your program as Q2b.c

Part a

```c
#include <stdio.h>

int main(){

    int i, j, userRating[5];
    int count1 = 0, count2 = 0, count3 = 0, count4 = 0, count5 = 0;

    while(1){
        printf("Input rating(1-5): ");
        scanf("%d", &userRating[i]);

        if(userRating[i] == -99){
            break;
        }

        if(userRating[i] < 1 || userRating[i] > 5){
            userRating[i] = 0;
        }

        if (userRating[i] == 1){
            count1++;
        }
        else if (userRating[i] == 2){
            count2++;
        }
        else if (userRating[i] == 3){
            count3++;
        }
        else if (userRating[i] == 4){
            count4++;
        }
        else if (userRating[i] == 5){
            count5++;
        }

    }

        printf("\n\n1\t");

        for(j=0 ; j < count1; j++){
            printf("* ");
        }

        printf("\n2\t");

        for(j=0 ; j < count2; j++){
            printf("* ");
        }
        printf("\n3\t");

        for(j=0 ; j < count3; j++){
            printf("* ");
        }
        printf("\n4\t");

        for(j=0 ; j < count4; j++){
            printf("* ");
        }
        printf("\n5\t");

        for(j=0 ; j < count5; j++){
            printf("* ");
        }

        printf("\n");

    return 0;
}
```

Part b

```c
#include <stdio.h>
int main(){

    int image[4][4];
    int i, j;

    for(i=0; i<4; i++){
        printf("Values for row %d\n", i+1);
        for(j=0; j<4; j++){
            printf("    Values for row %d: ", j+1);
            scanf("%d", &image[i][j]);
        }
    }

    printf("\n");

    for(i=0; i<4; i++){
        for(j=0; j<4; j++){
            printf("%d ", image[i][j]);
        }
        printf("\n");
    }

    printf("\n");

    for(i=0; i<4; i++){
        for(j=0; j<4; j++){
            if(image[i][j] < 55){
                image[i][j] = 0;
            }
            else{
                image[i][j] = 1;
            }

            printf("%d ", image[i][j]);
        }
        printf("\n");
    }
    printf("\n");

    return 0;
}
```

2023- session1

PART A

Write a C program to do the following.

a) Declare a 1D array called **strArr** of size 15.
b) Input a text from the keyboard and store in the array.
c) Reverse and store the elements of the array **without using an additional array**.
d) Display the array elements.

Hint: swap first and the last element, second and one before the last, etc…

Example

Original Array

L	O	V	E	P	R	O	G	R	A	M	M	I	N	G

Reversed Array

G	N	I	M	M	A	R	G	O	R	P	E	V	O	L

Save your program as **arrayB1.c**

PART B

Write a C program to do the following

a) Declare a 2D array called **salesArr** with 2 rows and 3 columns. Here the rows represent the salesperson and column represent the products.
b) Input the sales amount of 2 salesperson for 3 products and store in the array.
c) Display the sales amount in the tabular format representing salesperson as rows and product as columns.
d) Find the index of the salesperson with the highest total sales amount.

Save your program as **arrayB2.c**

Sample program output

```
Enter the sales amount of sales person 1.
                Product 1: 2000.00
                Product 2: 4500.00
                Product 3: 1200.00
Enter the sales amount of sales person 2.
                Product 1: 3500.00
                Product 2: 2100.00
                Product 3: 2300.00
```

The sales amount of 2 sales person for 3 products.

```
2000.00     4500.00     1200.00
3500.00     2100.00     2300.00
```

The sales person with the highest sales amount is sales person **1**

Part a

```c
#include <stdio.h>

int main(){

    char strArr[15];
    int i;

    for(i=0; i<15; i++){
        printf("Input character %d: ", i+1);
        scanf(" %c", &strArr[i]);
    }

    printf("\n");
    printf("Original Array\n");

    for(i=0; i<15; i++){
        printf("%c ", strArr[i]);
    }

    printf("\n\nReversed Array\n");

    for(i=15; i>=0 ; i--){
        printf("%c ", strArr[i]);
    }

    return 0;
}
```

Part b

```c
#include <stdio.h>

int main(){

    float salesArr[2][3];
    float total = 0, max = 0;
    int count;
    int i, j;

    for(i=0; i<2; i++){
        printf("Enter sales amount of sales person %d.\n", i+1);
        for(j=0; j<3; j++){
            printf("    Product %d: ", j+1);
            scanf("%f", &salesArr[i][j]);

            total = total + salesArr[i][j];

        }
            if(max < total){
                max = total;
                count = i+1;
            }
            total = 0;
    }

    printf("\n\nThe sales amount of 2 sales person for 3 products.\n");

    printf("\n");

    for(i=0; i<2; i++){
        for(j=0; j<3; j++){
            printf("%.2f\t\t", salesArr[i][j]);
        }
        printf("\n");
    }
    printf("\n");

    printf("\nThe sales with the highest sales amount is sales person %d\n", count);

    return 0;
}
```

2023- session2

Question 3 **30 Marks**

PART A

Write a C program to do the following.

a) Declare a 1D array called **rainArr** of size 7.
b) Input the rainfall of 7 days from the keyboard and store in the array.
c) Read the values stored in the array, calculate and display the average, maximum, minimum rainfall of the values for day 1 to 1, day 1 to 2, day 1 to 3, ...

Example

45.0	60.0					

Input the rainfall of day 1: 45.0

Input the rainfall of day 2: 60.0

.....
.....

Day 1 to 1

Average rainfall : 45.00 mm
Maximum rainfall : 45.0 mm
Minimum rainfall : 45.0 mm

Day 1 to 2

Average rainfall : 52.50mm
Maximum rainfall : 60.0 mm
Minimum rainfall : 45.0 mm

.....
.....

Save your program as **arrayD1.c**

PART B

Symmetric matrix is a square matrix in which elements in either side of the main diagonal are **equal**

e.g.

8	4	1	9
4	2	8	7
1	8	1	3
9	7	3	1

a) Create a two dimensional integer array called **matrixS** of size 4 x 4 and allow the user to enter values from the keyboard.

User interface should be as follows:

Values for row1
Enter element 1 :
Enter element 2 :
Enter element 3 :
Enter element 4 :
Values for row2
Enter element 1 :
Enter element 2 :
Enter element 3 :
Enter element 4 :
.....
.....

b) Display the contents of **matrixS** array and check whether it is a symmetric matrix or not and display an appropriate message.

Save your program as **arrayD2.c**

Part a

```c
#include <stdio.h>

int main(){

    float rainArr[7];
    int i;
    float average,  max = 0, min = 1e9 , k = 1.0, total = 0;


    for(i=0; i<7; i++){
        printf("Input the of day %d: ", i+1);
        scanf("%f", &rainArr[i]);
    }

    printf("\n");

    for(i=0; i<7; i++){
        printf("%.1f ",rainArr[i]);
    }

    printf("\n\n");

    for(i=0; i<7; i++){
        printf("Day 1 to %d\n", i+1);

        total = total + rainArr[i];
        average = total / k ;

            if(max < rainArr[i] ){
                max = rainArr[i] ;
            }
            if(rainArr[0] >= rainArr[i] ){
                min = rainArr[i] ;
            }

        printf("Average rainfall: %.2f mm\n", average);
        printf("Maximum rainfall: %.2f mm\n", max);
        printf("Minimum rainfall: %.2f mm\n\n", min);

        k++;
    }

    return 0;
}
```

Part b

```c
#include <stdio.h>

int main(){

    int matrixS[4][4];
    int i, j, symmetricMatrix = 1;


    for(i=0; i<4 ; i++){
        printf("Values for row %d\n", i+1);
        for(j=0; j<4; j++){
            printf("\tEnter element %d: ", j+1);
            scanf("%d", &matrixS[i][j]);
        }
    }

    printf("\n");

    for(i=0; i<4 ; i++){
        for(j=0; j<4; j++){
            printf("%d ", matrixS[i][j]);
        }
        printf("\n");
    }

    if(matrixS[1][0] == matrixS[0][1] && matrixS[2][0] == matrixS[0][2] && matrixS[3][0] == matrixS[0][3]
    && matrixS[3][1] == matrixS[1][3] && matrixS[2][1] == matrixS[1][2] && matrixS[3][2] == matrixS[2][3])
        printf("\nThis is symmetric matrix.\n");
    else{
        printf("\nThis is not a symmetric matrix.\n");
    }
    printf("\n");

    return 0;
}
```

Model paper Question

Question 2

Smart meter reading is an electronic device that records consumption of electric energy in intervals of an hour or less and communicates that information at least daily back to the utility system for monitoring and billing.
Write a C program to store the daily electricity usage of a month (assume that one month has 28 days and exactly 4 weeks) and generate a summary report.

a) Use a two dimensional array to store the electricity usages. One row in the array should store the usage of a week.

   Following are the electricity usage of a house in Victoria in the month of November. Store these data in the array.

   1.2, 2.1, 0.8, 0.0, 1.0, 1.3, 4.0, 1.5, 3.2, 1.0, 1.3, 4.0, 1.5, 3.2, 2.3, 0.4, 1.2, 2.1, 0.8, 0.0, 2.6, 2.1, 1.7, 7.0, 1.2, 2.1, 0.8, 0.0

b) Generate the summary report as below based on the values stored in the array.

   Day of the Month with maximum usage:
   Maximum Usage in units:
   Day of the Month with minimum usage:
   Minimum Usage in units:
   Average usage for the Month:
   Total usage for the Month:

```c
#include <stdio.h>

int main(){

    int i, j, m, n;
    float total = 0, min = 1e9, average = 0, max = 0, usage[4][7];

    for(i=0; i<4; i++){
        printf("Week %d\n", i+1);

        for(j=0; j<7; j++){
            printf("Electricity usage of day %d: ", j+1);
            scanf("%f",&usage[i][j]);

            total = total + usage[i][j];

            if (max < usage[i][j]){
                max = usage[i][j];
                m = j+1;
            }
            if(min > usage[i][j]){
                min = usage[i][j];
                n = j+1;
            }
        }
        average = total / 28.0;

        printf("\n");
    }

    printf("\n");

    for(i=0; i<4 ; i++){
        for(j=0; j<7; j++){
            printf("%.1f  ",usage[i][j]);
        }
        printf("\n");
    }

    printf("\nDay of the month with maximum usage: %d\n", m);
    printf("Maximum usage in units: %.2f\n", max);
    printf("Day of the month with minimun usage: %d\n", n);
    printf("Minimum usage in units: %.2f\n", min);
    printf("Total usage for the month: %.2f\n", total);
    printf("Average usage for the month: %.2f\n", average);

    return 0;
}
```