Question 1

An array VAL[1..15][1..10] is stored in the memory, with each element requiring 4 bytes of storage. If the base address of array VAL is 1500, determine the location of VAL[12][9] when the array VAL is stored row-wise.

Question 2

Given an anti-diagonal matrix with dimensions of M[100][100]. Then, the value of M[i][j] is stored in _____ location of the 1-D array if the elements of the 1-D array are only anti-diagonal elements. Assume array indexing starts from 0. (Hint: An anti-diagonal matrix is a matrix where all the entries are zero except those on the diagonal going from the lower left corner to the upper right corner)

Question 3

Find the output of the C program below, assuming the base address of the array is 3238170320. The size of int is 4 bytes.

```
int main()
{
    Int a[3][3] = {{3,2,1}, {6,5,4}};
    printf("%u, %u, %u\n", a, a+1, a+3);
    return 0;
}
```

Question 4

Consider an array A[1…...17][1…….17], given that the array is to be stored in column-major order. Address of A[12][15] is 2048 and that of A[1][16] is 2066. Find the size of each element of the array. Also, find the address of the base element.

Question 5

Given an integer array arr[0...99] [0…..99], with base address as 1024. A user decides to fill the data from arr[15][0] instead of arr[0][0]. So he fills the entries in arr[15][0] , arr[15][1] , arr[15][2]..... arr[15] [99] , then again from arr[16][0] , arr[16][1] and so on. Given that the size of an integer is 4 Bytes, row-major ordering is used. What is the address of the 121st entry(entries are counted as 1,2,3, and so on).

Question 6

Consider an array defined as M[0…..10][0…..20]. When we refer to an array element M[P][Q], we wish to check that this is a valid access to an array element. Consider the following tests for the validity of the array reference

I. The address of M[P][Q] is an address within the range of memory allocated for the array M

II. $0 \leq P$ and $0 \leq Q$

III. $P \leq 10$ and $Q \leq 20$

Which combinations of the above tests will minimally guarantee that the array access is valid?

Question 7

What is the output of this program?

```
int main()
{
        int m[10] = {0};
        int x = 0;
        m[x] = ++x;
        printf("%d %d %d", m[0], m[1], m[2]);
        return 0;
}
```

Question 8

An nxn matrix R where n ranges from 1 to n is defined as follows:

$R[i,j] = i$ if $(i==j)$

$R[i,j] = i^2-j^2$ if $(i<j)$

$R[i,j] = j^2-i^2$ if $(i>j)$

The sum of the elements of array R is ……

Question 9

Consider a 2D array A with its elements stored in the form of a lower triangular matrix. The elements must be crossed to read A[4][2]. The indexing of the array can be represented as A[-6,.....,+8][-6,.....,+8]. The base address of the array is 1000. The elements are stored in row-major order. The address to access the element A[4][2] is ……

Question 10

Consider a 3D array A[90][30][40] stored in column-major order. The base address starts at 10. The location of A[20][20][30] is _____ [Assume the first element is stored in A[1][1][1] and each element takes 4 Bytes].

Question 11

Consider the following statements about 1D array in C:

I. Arrays are stored in contiguous locations in memory
II. Arrays allow the users to random access any location of the array
III. A single array can have elements of different data types
IV. An array is a homogeneous data structure

Which of the above statements are True?

Question 12

How do you initialize a 1D array in C?

A:int arr[3] = {1,2,3};
B:int arr[3] = (1,2,3);
C:int arr[3] = [1,2,3];
D:int arr(3) = (1,2,3);

Question 13

What is the default value of elements in an uninitialized array in C?

A:Garbage
B:NULL
C:0
D:It depends on compiler

Question 14

Consider the C code snippet below:

```
#include <stdio.h>
void f(int a[2][])
{
    a[0][1] = 3;
    int i = 0, j = 0;
    for (i = 0;i < 2; i++)
    for (j = 0;j < 3; j++)
    printf("%d", a[i][j]);
```

```
}
void main()
{
    int a[2][3] = {0};
    f(a);
}
```

A:Compile time error
B:Runtime error
C:All 0s
D:030000

Question 15

Consider the C code snippet below:

```
int main()
{
   int A[5] = {1,3,5,7,9};
   printf("%d", A[6]);
}
```

What happens after the execution of the above code?

Question 16

Consider the C code snippet below:

```
int main()
{
   int A[2][4] = {{1,3,5,7}, {1,2,9}};
   printf("%d", A[1][3]);
}
```

What happens after the execution of the above code?

Question 17

What does the function func(int A[][N], int B[][N]) do?

```
void func(int A[][N], int B[][N])
{
    int i, j;
    for (i = 0; i < N; i++)
        for (j = 0; j < N; j++)
            B[i][j] = A[j][i];
}
```

Question 18

Consider the function func(int arr[], int size) implemented to reverse an array. Complete the function by filling in the blank.

```
void func(int arr[], int size)
{
    int Arr[size];
    for (int i = 0; i < size; i++)
   {
        S1:_____
    }


    printf("Reversed Array: ");
    for (int i = 0; i < size; i++) {
        printf("%d ", Arr[i]);
    }
}
```

Question 19

Consider the following function func(int arr[], int d, int n) to left rotate an array by d elements.
For example,
arr[] = {1, 2, 3, 4, 5, 6, 7}, d = 2.
Left Rotation is done by 2 times.
So the array becomes arr[] = {3, 4, 5, 6, 7, 1, 2}

Complete the function by filling in the blanks.

```
void func(int arr[], int d, int n)
{
    int p = 1;
    while (p <= d) {
        int last = arr[0];
        for (int i = 0; i < n - 1; i++) {
          S1:____________________
          }
          S2:_____________________
        p++;
    }
}
```

Question 20 - multiple choice

In a compact single dimensional array representation for lower triangular matrices (i.e, all the elements above the diagonal are zero) of size n × n, non-zero elements (i.e., elements of the lower triangle) of each row are stored one after another. Starting from the first row, the index of the (i, j)th element of the lower triangular matrix in this new representation is:……..