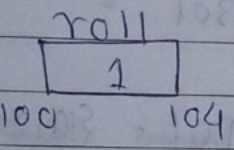
Assignment - 4

PAGE No.
DATE / /

1) Read below statements and draw its diagrammatic layout.

```
int roll = 1;
Const int No = 10;
Const Char CH = 'A';
Const int arr[6] = { 10, 20, 30, 40, 50, 60 };
```

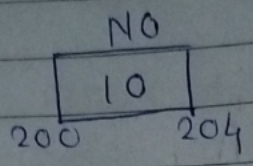
```
int roll = 1
```



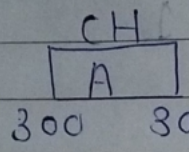

```
#include <stdio.h>
int main ()
{
    int roll = 1;
    Const int No = 10;
    Const Char CH = 'A';
    Const int arr[6] = {10, 20, 30, 40, 50, 60};

    return 0;
}
```

PAGE No.
DATE / /

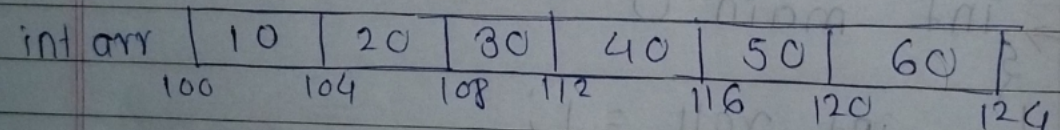
const int NO = 10;



const char CH = 'A';



const int arr[6] = {10, 20, 30, 40, 50, 60};



PAGE No.
DATE / /

2] Explain the Concept of the one dimensional array with example, diagrammatic layout.

→ Array:-

- Array is considered as linear data structure

- Array is derived datatype which holds multiple homoginous elements in index formal

- ~~There are two types of~~ Array is classified as

1] one dimensional Array

2] multidimiensional Array

3] 2D Array (2 dimensional Array)

1] One dimensional Array

- one dimensional Array is one type of the Array it holds homogineous elements.

- one dimensional array consists of a fixed number of elements of the same data type organized as a simple linear sequence.

- The elements of a single or one dimensional array can be accessed by using a single subscript operator.

Declaration of one dimensional Array

Data type Array Name [length of Array] = {data elements};

Subscript operator

for example:

```
int Arr[4] = {10, 20, 30, 40};
```

PAGE No.
DATE / /

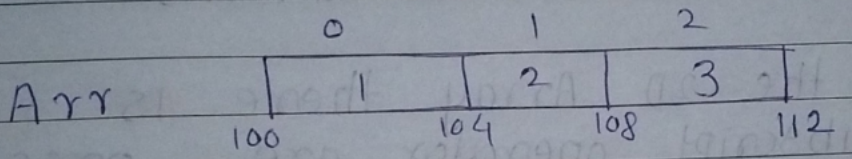
For example:

```
# include <stdio.h>
int main ()
{
    int Arr[3] = {1, 2, 3};
    printf(" elements in Array \n", Arr[3])
    return 0;
}
```

OS → ① → int main () ←

② ↓ int Arr[3] = {1, 2, 3};
printf(" elements in Array \n", Arr[3])

return 0; ← ③



sizeof (Arr) = 12

3] Explain the concept of 2D Array with example and diagrammatic layout

→ 2D (two dimensional Array)

- 2D Array is the one type of the Array which store homongenious values or elements

- In the two dimensional array contains th one dimensional array.

- In the 2D Array there is two subscript operator are present

- the first subscript operator in the 2D Array is for the row and the second subscript operator is for the column.

- we can intialize 2D Array using multiple ways.

for example:-

Arr [3] [3] = {1, 2, 3, 4, 5, 6, 7, 8, 9

for example:

```
#include <stdio.h>
int main()
{
    int Brr[3][3] = {1,2,3,4,5,6,7,8,9};
    Printf("Elements in 2D Array\n", Brr[3][3]);
    return 0;
}
```

OS ① ② ③

Dry run

row column

Brr [3] [3]

ROW \ column	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

Brr[Row] [column]

	column 0	1	2
0	[0] [0]	[0] [1]	[0] [2]
1	[1] [0]	[1] [1]	[1] [2]
2	[2] [0]	[2] [1]	[2] [2]

Brr [2] [1] = 8

4] Write a program to display the content of below array, draw it's diagrammatic layout

const int arr[6] = {10, 20, 30, 40, 50, 60};

→

```
#include <stdio.h>
int main()
{
    int arr[6] = {10, 20, 30, 40, 50, 60};
    printf("Elements in Array are \n", arr[6]);
    return 0;
}
```

OS → ① → ② → ③

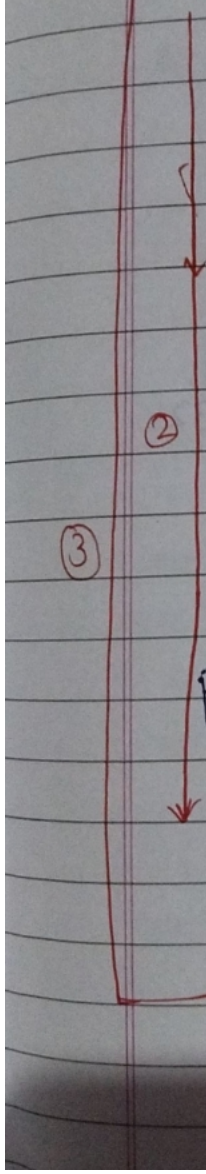
Dry run:-

	0	1	2	3	4	5
arr	10	20	30	40	50	60
	100	104	108	112	116	120	124

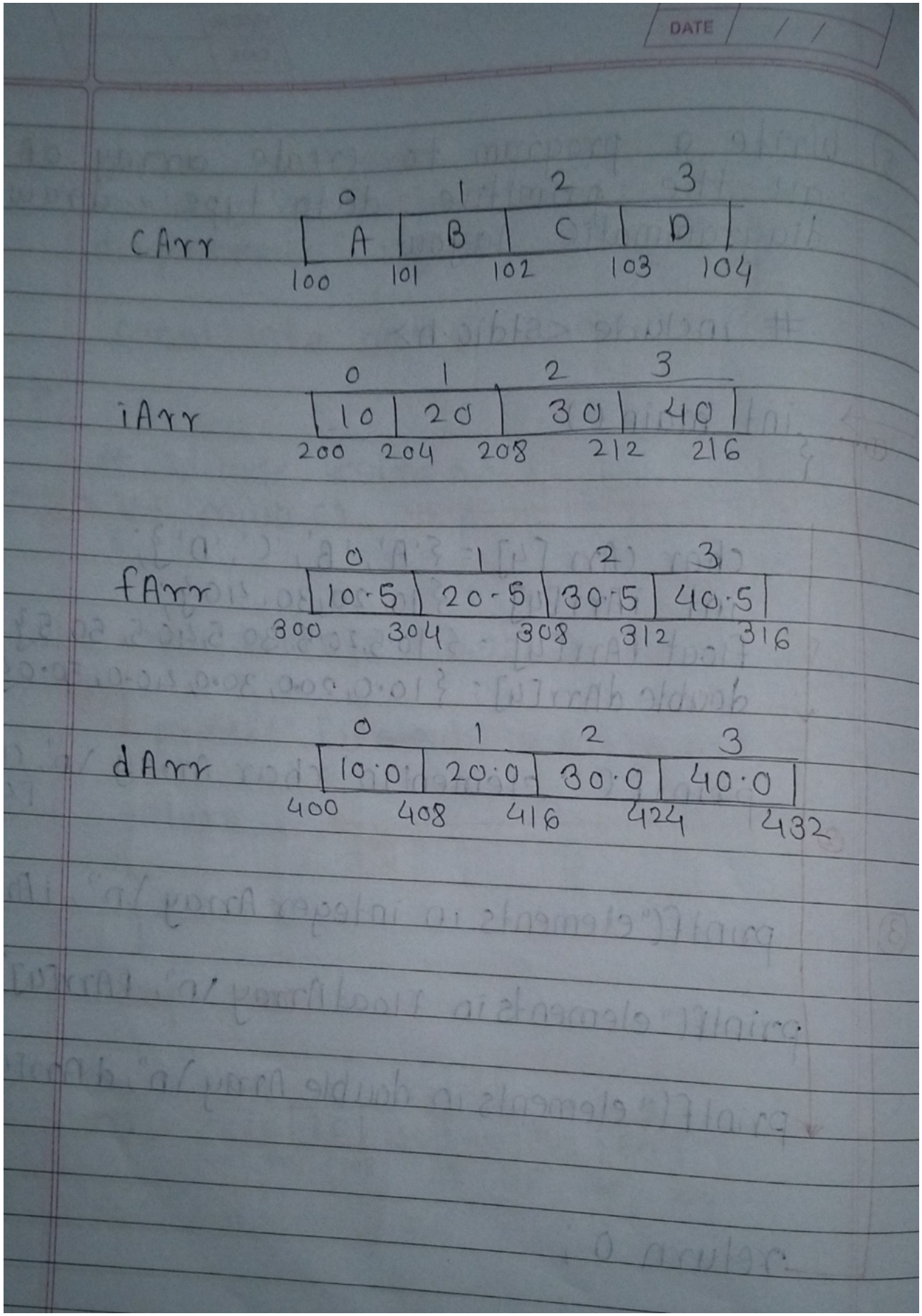
5] Write a program to create array of all the primitive data type, draw diagrammatic layout.

→

```
#include <stdio.h>

int main()
{
    char cArr[4] = {'A', 'B', 'C', 'D'};
    int iArr[4] = {10, 20, 30, 40};
    float fArr[4] = {10.5, 20.5, 30.5, 40.5, 50.5};
    double dArr[4] = {10.0, 20.0, 30.0, 40.0, 50.0};

    printf("element in char Array \n", cArr[4]);

    printf("elements in integer Array \n", iArr[4]);

    printf("elements in float Array \n", fArr[4]);

    printf("elements in double Array \n", dArr[4]);

    return 0;
}
```



DATE / /


cArr
0 1 2 3
A B C D
100 101 102 103 104
iArr
0 1 2 3
10 20 30 40
200 204 208 212 216
fArr
0 1 2 3
10.5 20.5 30.5 40.5
300 304 308 312 316
dArr
0 1 2 3
10.0 20.0 30.0 40.0
400 408 416 424 432

6] Explain the different approaches (ways) to create the array in c/c++

→ Array:-

- Array is Derived datatype which holds multiple homogenious elements in index formal
- Array is considered as a linear data structure

followings are the ways to create the array in c/c++

1) With size and initalization list.

we can create the Array by specifying the length of array as well as initalize the array immediately using member initalization list.

- If the array elements are initalized inside { } bracket is called member initalization list

Syntax:-

```
Datatype ArrayName [length]
    = { data elements } ;
```

for example:

int Arr [5] = { 10, 20, 30, 40, 50 };

2] Array initalization with declaration and withaut length.

- At the time of creating Array if we initalized it using member initalization list then it is optional to specify the length of array in [] because the length of Array is Automatically deduced / calculated by Compiler.

Syntax:

Datatype ArrayName [] = { data element };

for example:

int Brr [] = { 10, 20, 30 };

PAGE No.
DATE / /

3) Declaration and but initalization with less number

- At the time of creating Array if we initalized it using member initalization list then it is optional to specify the length of Array is in [] because the length of Array is Automatically

3) Declaration and but initalization but with less number

- It is also possible to provide the length of array with some value but initilization less no of elements in il.

for example:-

```
int arr[7] = {10,20,30,40};
```

- If we initalize less no of elements then all remain memory no of Array get Automatic initalize array of default and its depend on datatype of Array.

4] We can also create the Array & initalize it by using the concept of member by member initalization (initialization after declaration)

for example

```
int Arr[3];
Arr[0] = 10;
Arr[1] = 20;
Arr[2] = 30;
```

5] We can create the Array of constant data element.

- when we create the Array which contain constant we cannot, increment decrement any value from that array.

- If we want to create Constant Array then the members of that must be initalization using member initalization list only.

for example:

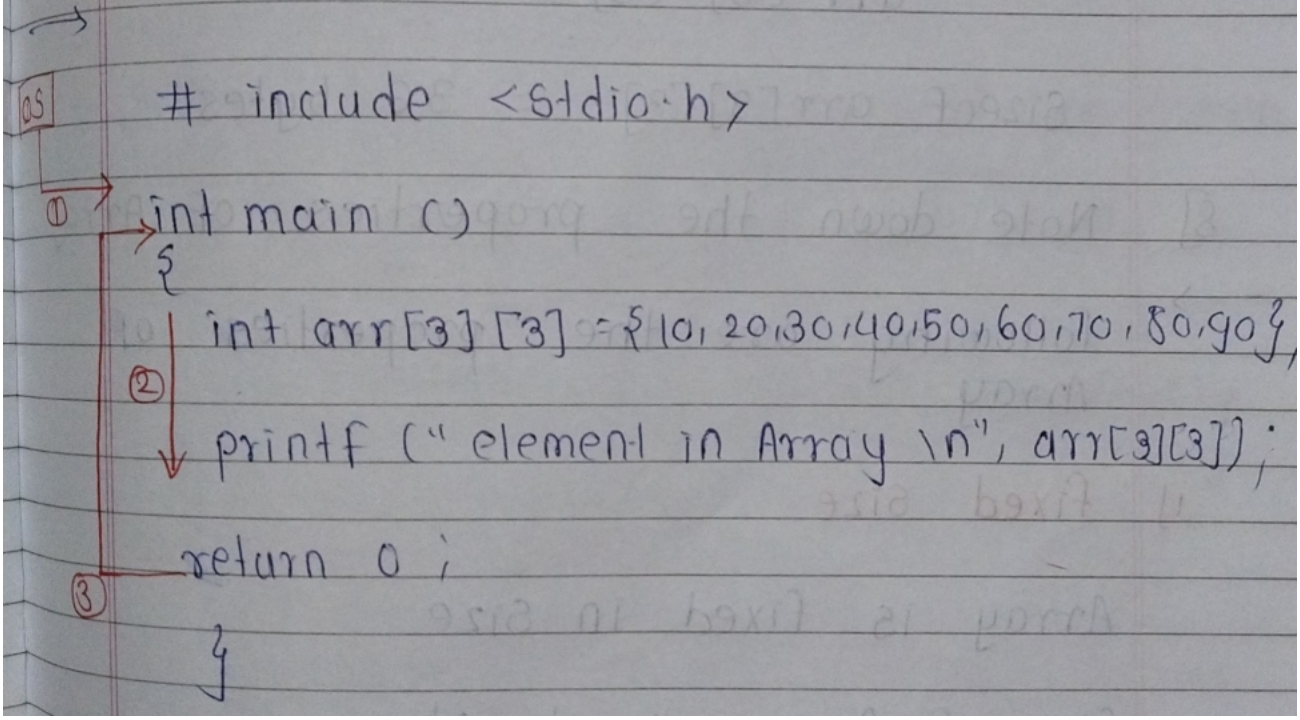
```
const in Arr[4] = {1,2,3,4};
```

6) when we consider the name of array then it is consider as the base address (1 value) of the first element of that Array

In case of Array there is no need use of and to fetch the address

7) Draw the diagrammatic layout of and the matrix of below array.

int arr[3][3] = {10, 20, 30, 40, 50, 60, 70, 80, 90};



Matrix of Array

arr[3][3]

	0	1	2
0	10	20	30
1	40	50	60
2	70	80	90

Row \ Column	0	1	2
0	[0][0]	[0][1]	[0][2]
1	[1][0]	[1][1]	[1][2]
2	[2][0]	[2][1]	[2][2]

arr [3] [3]

Sizeof arr[3][3] = 36 bytes

8] Note down the properties of Array

→ following are the properties of Array

1] fixed size

Array is fixed in size

- Size of Array can't change

PAGE No.
DATE / /

2) Homogenious Collection

- Array can store only one type of data element

3) Array indexing

- Array index always start from 0
- last element ~~will~~ be in Array will be n-1
- where n is length of Array

4) Dimensions of Array

Array has one dimensional, two-dimensional & also multidimensional

5] Continous storage

- Array elements are stored continously one after another in the memory

6] Random Access

Array provides random access to its elements.

7] No checking out of index Bound

In C, C++ there is no provision for index out of bound checking.

- It can compile program but produces unexpected output at runtime

8] Length of Array

- In C, C++ program there is no inbuild function to find the length of Array but we can do it by using

1] using Sizeof() operator

2] using pointer Arithmatic

3] using loops.

PAGE No.
DATE / /

9] Explain the Concept of length of Array, Size of Array

→ length of Array

- Array is derived data type in C programming language
- By using [] Subscript operator we can create the Array of any datatype.
- In the C, C++ language there is no inbuilt function for calculating the length of Array.
- the length of Array is given in the subscript operator when the length is not given then we calculate the length of Array by using below methods.

1) Sizeof () operator

e.g: int length of Array = (Sizeof (Arr) / Sizeof (Arr[2]);

2) pointer Arithmatic
3) using loops.

DATE / /

Size of Array

- By using the length of Array and which datatype is used in that Array we can calculate the size of Array.

Size of Array = length of Array * datatype of Array

for example:

```
int Arr [2] = {1, 2};
```

Size of Arr[2] = 2 * datatype of Array

= 2 * 4

Size of Arr [2] = 8

PAGE No.
DATE / /

10] Write a notes on Datatype modifiers and Datatype qualifiers.

Datatype modifier

- Data type modifier the base type to yield a new type.
- In Datatyp modifier can modify or change data type
- Datatype modifier classify as

1) Sign

2) Size

1) Sign modifier

- Sign modifiers will affect the ~~size~~ of sign of the variabe or data object
- two types of sign modifier

1) signed modifier

- If we create variable withoud specifying signed or unsigned keyword thad

considered as signed variable.

- Signed modifier variable stor any positive, negative, 0 value

2) unsigned modifier

- This modifier should be written variable name if we want to store only positive and 0 value in it.

example: unsigned int a = 40;

2) Size modifier

- The size modifier will affect on the size of data object.

- There are 2 types of size modifier

1) Short size modifier

- If we use short keyword with integer datatype we get 2 bytes of memory

2) long size modifier

If we use long keyword with integer datatype we get 8 bytes memory

PAGE No.
DATE / /

Data type Qualifier

- Data type qualifier used to indicate the special properties of data within an object.
- two types of Data Qualifier :-

1) constantqualifier

- If we declare the ~~ob~~ data object as a constant then we cannot change the value during the execution of a program.

e.g: Pi = 3.14;

2) volatile Qualifier

- volatile qualifier annouces that the data object has some special properties relvant to optimization.