

Assignment]

Q-1. Explain 2-D Array in Detail in C

* Definition:

→ When the list with variable is stored using two subscripts (rows and columns), then it is known as two dimensional array.

→ Syntax

Data-type array-name[rows][columns];

→ Here row indicates the maximum numbers of horizontal elements and column indicates the maximum numbers of vertical elements.

→ First array element is stored in array-name[0][0].

→ Ex int number[~~3,3~~][2,3];

→ if numbers are 5, 10, 15, 20, 25, 30 then the values are stored as

	0	1	2	
0	0,0	0,1	0,2	5 10 15
1	1,0	1,1	1,2	20 25 30

$$\therefore \text{numbers}[0][0] = 5$$

$$" \quad [0][1] = 10$$

$$" \quad [0][2] = 15$$

$$" \quad [1][0] = 20$$

$$" \quad [1][1] = 25$$

$$" \quad [1][2] = 30$$

→ Initialization:-

① Compile time:

→ Example:

```
int a[2][2] = {{1,2},  
                {3,4}};
```

② Run time:

→ Example:

```
int a[2][2], i, j;  
for (i=0; i<2; i++)  
{  
    for (j=0; j<2; j++)  
    {  
        a[i][j]=0;  
    }  
}
```

* Char type 2-D Array

→ 2D character arrays are very similar to the 2D integer arrays

→ A 2D character array is more like a String array. It allocates us to store multiple strings under the same name.

→ String is nothing but an array of characters which ends with a '\0'.

→ Declaration for char type 2-D array.

Syntax :

char variable_name[subscript1][subscript2];

char a2[5][20];

→ Initialization

Ex:-

char name[5][10] = {"tree", "bowl", "hat", "mice", "toon"};

Memory location	Array elements									
25860	t	r	e	e	\0					
25870	b	o	w	l	\0					
25880	h	u	t	\0						
25890	m	i	c	e	\0					
25900	f	o	s	n	\n					

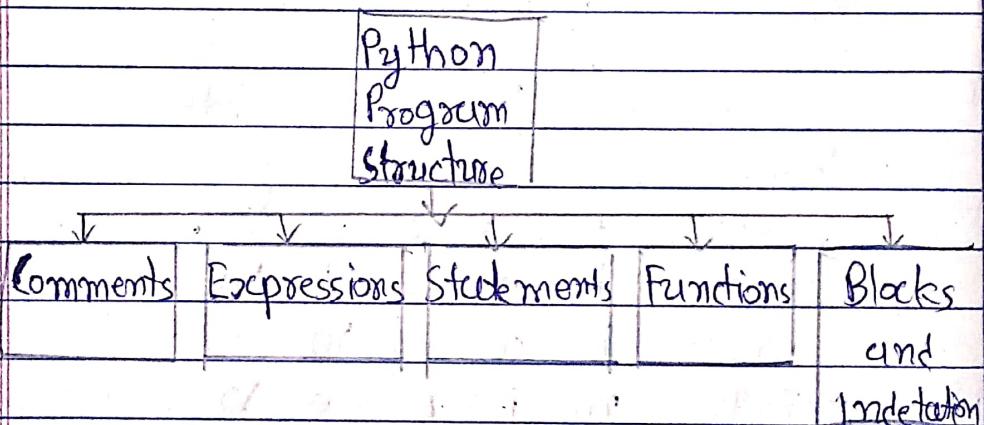
→ length of each string is 10

→ and we stored 5 string in different memory location

→ Each character occupies 1 byte of storage from the memory.

2. Explain Structure of Python Programming Language.

- An interpreter directly executes instructions line by line written in a programming or scripting language without converting them to an object code or machine code.
- Examples of interpreted languages are Perl, Python and Matlab.
- Basic structure of python programming includes following components:



- Comments
 - Comments are the additional readable information to get better understanding about the source code.
 - comments in Python are the non-executable statements
 - ① Single Line Comment → `# Single line comment`
 - ② Multiline Comment
`''' Simple program
of python '''`

→ Expressions:

- An expression is any legal combination of symbols that represents a value.
- An expression represents something which python executes and which produces a value
- Eg. 10, $x + 5$

→ Statements:

- A statement is a programming instruction that does something i.e., some action takes place
- A statement executes and may or may not results in a value
- E.g. `print(x+2)`, $y = x + 5$, `x = 10`

→ Functions:

- A function is a code that has a name and it can be reused (executed again) by specifying its name in the program, where needed.
- A function begin with 'def' statement
- Eg. `goingsgood()`

→ Block and Indentation

- A group of statements which are part of another statement or a function are called block or code-block or suite in python.
- Indentation is used to show blocks in python.
Four spaces together mark the next indent level

A simple python Program!

Comments → `# This is a simple program`
function → `# Definition of function goinggood()`
defined → `def goinggood():`
 `print("Python")`
 `# Main Program Code`

statement → `x = 10` ← Expressions
 `y = x + 5` ←
 `print(x)` : inline comment

if `y < 25:` # colon mean it's block
 `print("initial value of x is < 10")`

block → `else:`
 `print("initial value of x is ≥ 10")`

Function → `goinggood()`
called →

→ Python code indentation and execution

- Indentation refers to the spaces at the beginning of code line.
- While in other programming languages the indentation in code is for readability only, the indentation in python is very important.
- Python uses indentation to indicate a block of code.
- The leading whitespace (space and tabs) at the start of a line is used to determine the indentation level of the line.

- increase the indent level to group the statements for that code block. Similarly, reduce indentation to close the grouping
- Example

```
def foo():           first level indentation  
    print("Hi")   → foo() method statements.  
  
    if True:  
        print("true") 2nd level indentation  
    else:  
        print("false")  
  
print("Done") ← code without indentation
```

→ Python Indentation Rules:

- We can't split indentation into multiple lines using backslash.
- The first line of Python code can't have indentation, it will throw indentation Error.
- You should avoid mixing tabs and whitespaces to create indentation.
- It is preferred to use whitespaces for indentation than the tab character.
- The best practice is to use 4 whitespaces for first indentation and then keep adding additional 4 whitespaces to increase the indentation.

3. Explain Element address in Array
 • (Row Major and Column Major):

→ Row Major Ordering in an Array:

→ Row major ordering assigns successive elements, moving across the rows and then down the columns, to successive memory locations

Memory				
0	1	2	3	8 A[2,2]
0	0	1	2	7 A[2,1]
1	3	4	5	6 A[1,2]
2	6	7	8	5 A[1,1]
→				4 A[2,0]
↓				3 A[1,0]
↓				2 A[0,2]
↓				1 A[0,1]
↓				0 A[0,0]

→ The formula to compute the Address for a two-dimension row-major ordered array as:

$$\text{Address of } A[i][j] = \text{Base Address} + w \times (C \times j + i)$$

Where,

Base Address is the address of the first element in an array.

w = is the width(size) of a data type

C = is total no of columns

- $i =$ is the row number of an element whose address is to find out
- $j =$ is the column numbers of an element whose address is to find out

Example:

- A matrix $P[15][10]$ is stored with each element requiring 8 bytes of storage. If the base address of $P[0][0]$ is 1400.
- Determine the address of $P[10][7]$ when the matrix is stored in Row major wise.
- Address of $A[i][j] = \text{Base Address} + w \times (c \times j + i)$
$$= 1400 + 8 \times (10 \times 10 + 7)$$

$$= 1400 + 858$$

$$= 2256$$

→ Column Major Ordering.

- If the element of an array is being stored in column wise fashion then it is called column major ordering in an array.
- Pictorially, a column-major ordered array is organized as shown below

0	1	2		$A[2, 2]$
0	1	2		$A[1, 2]$
1	3	4	5	$A[0, 2]$
2	6	7	8	$A[2, 1]$
				$A[1, 1]$
				$A[0, 1]$
				$A[2, 0]$
				$A[1, 0]$
				$A[0, 0]$
memory				

The formula is as follows:

$$\text{Address } A[i][j] = \text{Base Add} + W \times (R \times J + I)$$

Where,

Base add is the address of first element in
an array.

W = is the weight(size) of a data type

R = is Total no of Rows

I = is the Row Number of an element
whose address is to find out.

J = is column number of an element
whose address is to find out.

→ Example:

A matrix $P[25][20]$ is stored with each element
requiring 8 bytes of storage.

→ If the base address at $P[0][0]$ is 1500.

Determine the address at $P[10][7]$ when the
matrix is stored in Column Major wise.

$$= \text{Base Add} + W \times (R \times J + I)$$

$$= 1500 + 8 \times (15 \times 7 + 10)$$

$$= 1500 + 920$$

$$= 2420$$

Q. 4. Differentiate between Structure and Union.

	Structure	Union
key-word	Structure is used to define structure	Union is used to define a union
size	When variable is associated with a structure the compiler allocates the memory for each member. The size of structure is global and greater than or equal to the sum of size of its members.	When a variable is associated with union, the compiler allocates the memory by considering the size of the largest memory. So size of union is equal to the size of largest member.
Memory location	Each member within a structure is assigned unique storage area of memory.	Memory allocated is shared by individual members of union.
Value	Altering the value of a member will not affect other member of the structure.	Altering the value of any of the member will affect other members values.
Accessing	Individual members can be accessed at a time.	Only one member can be accessed at a time.
Initialization	Several members of a structure can initialize at once.	Only the first member of a union can be initialized.