constructor → used to allocate the memory to an obj. of the class.

C	C++
- structed prog. lang.	oopl.
- subset	superset
- data & func are free entities	- encapsulated
- doesnt support data hiding	does
- scanf() & printf() for i/p & o/p.	- cin & cout.

Reference	Pointer
- behaves like an alias for an existing var.	variable which stores the address of a variable
- null value can't be assigned	can be assigned
- necessary to initialize the var. at the time of declaration	x

Data Types:

int - 2, 4 bytes	double - 8
char - 1	short int - 2
float - 4	long double - 10 (12 or 16)

Tokens → keyword, identifier, constant & symbol.

OOPs →
1. Class → blueprint
2. Obj. → instance of class.
3. Inheritance → provides reusability & elimates redundancy of code.
4. Encapsulation → binds the data within a class
5. Abstraction → if members are defined with a public keyword, then the member are accessible outside also.

C++ access specifiers
1. Pvt : fun & var within same class
2. Pub
3. Protected → fun & var. cant be accessed outside the class except a child class.

used in (inheritance)

6. Data binding

7. Polymorphism →
- static (compile-time) → method overloading → diff functionality
- Dynamic (Run-time) → func overriding → child class & parent class → diff def. [child class overrides parent class fun.]

~~→ memor func~~ of class

Constructor → special method that initializes an object. (name must be same as class name)

Destructor → used to delete any extra resources allocated by the obj. (~)
→ called automatically once the obj. go out of scope.
→ same name & [no argument & no return type].

ML (SVM, KNN)

Call by Value :-

↳ Method copies the value of an arg. into the formal parameter of that func.

– In this parameter passing method, values of actual parameters are copied to func's formal parameters, & the parameters are stored in different memory locations.

⇒ So any changes made inside func's are not reflected in actual parameters of the caller.

Call by Reference :-

↳ Method copies the address of an argument into the formal parameter. In this method, the address is used to access the actual arg. used in the func call.

⇒ Memory allocation is the same as the actual parameters.

Bias :- error due to erroneous or overly simplistic assumptions in the lr. algo. you're using.

⇒ Model underfitting your data.

⇒ difficult to inc. ↑ accuracy.

Variance :- error due to too much complexity in the learning algorithm you're using.

⇒ highly sensitive to high degree of variation in Training data.

⇒ overfit data.

Supervised :- requires training labeled data.

eg. classification

~~KNN~~ ROC :- graphical represt. b/w true +ve rates & false +ve rates.

Precision :- known as +ve predictive value & it is a measure of amount of accurate +ve your model claims.

Recall :- known as true +ve rate &

⇒ the amount of +ve your model claims compared to the actual no. of +ve there are in your dataset.

L1 & L2 :

L2 regularization tends to spread error among all the terms, while L1 is more binary/sparse, with many variables either being assigned as 1 or 0 in weighting.

Python Questions :-

LIST	TUPLES
- Mutable	- Immutable
- slower	- faster
eg. ['a', 'b', 'c']	eg. ('a', 'b', 'c')

Key features of Python :-

1.) Interpreted language.

↓

does not need to be compiled before it is run

Source Code
↓
Intermediate language.
↓
Machine language
↓
Execution

2.) Dynamically typed.

↳ don't need to state the type of variables while declaring.

3.) well suited OOP.

4.) Doesn't have access specifiers. (public, pvt).

5.) writing code is quick

What type of lang. is Python?

capable of scripting, but in general sense, it is considered as a general purpose programming lang.

PEP8 :- Python Enhancement Proposal.

↳ set of rules : how to formate code (maximum readability)

Memory manged in Python

1.) Python private heap space.

↳ programmer doesn't have access to this pvt. heap. Python interpreter takes care of this instead.

2.) Allocation of heap space is done by Py's memory manager

3.) Py. also has inbuit garbage collector.

↳ recycles all the unused memory & so that it can be made available to the heap space.

name space in Python :-

Naming system used to make sure that names are unique to avoid naming conflicts.

PYTHONPATH

↳ environment variable which is used when a module is imported.

Python modules :-

files containing python code. This code can either be functions classes or variables.

Some built-in modules are:

1. OS.
2. SYS
3. math
4. random
5. data time
6. JSON

Global variables :-

Variables declared outside a func or in global space.

- These variables can be accessed by any func in the program.

Local variables :- Declared inside a func is known as L.V. This variable is present in the local space.

→ continuous memory loc.

Array Vs. List :-

↳ Both have the same way of storing data.

- Array - can holds on only a single data type elements whereas

list - can hold any datatype elements.

what is _init_ ?

↳ Method or constructor in python. This method is automatically called to allocate memory when a new object of a class is created

lambda function :-

- Anonymous function
- Can have any number of parameters but, can have just one statement.

a = lambda x,y : x+y

Self in Python :-

It is an instance or an object of a class. This is explicitly included as the first parameter.

- Helps to differentiate b/w the methods & attributes of a class with local variables.

Break, Continue & pass :-

Break - allows loop termination when some condition is met & the control goes to next statement

Continue - allows skipping some part of a loop when some specific condition is transferred. to the begining of the loop.

Pass - Used when you need some block of code syntactically, but you want to skip its execution.

↳ Null operation.

↳ Nothing happens when this executed.

Python iterators :-

Iterators are objects which can be traversed through or iterated upon.

Pickling & unpickling :-

- Pickle module accepts any Python object & converts it into a string representation & dumps it into a file by using dump function. This process is called pickling.

while the process of retrieving original python obj. from the stored string representation is called unpickling.

Generators :-

fun^c that return an iterable set of items. ~~are~~

docstrings in Python :-

↳ They are not actually comments, but they are documentation string.

- Triple quotes.
- serves the purpose of comments as well.

```
"""
 Python
"""
```

Shallow copy :-

Used when a new instance type gets created & it keeps the values that are copied in the new instance.

- It is used to copy the reference pointers just like it copies the values.
- These references point to the original obj. & the changes made in any member of the class will also affect the original copy of it.

⇒ Allows faster execution of the program & it depends on the size of the data that is used.

~~Shallow Copy~~ :-

Deep copy :-

- Used to store the values that are already copied.
- Doesn't copy the reference pointers to the objects.

⇒ The changes made in the original copy won't affect any other copy that uses the object.

⇒ Execution of prog - slower due to making certain copies for each obj. that is being called

*args, **kwargs :-

- we use *args when we aren't sure how many arguments are going to be passed to a function,

**kwargs is used when we don't how many keyword arguments will be passed to a function, or it can be used to pass the values of a dictionary as keyword arguments.

- These both arguments are conventional, we could used any other name as well. (*bob, **billy)

Multidimensional array :-

↳ defined as array of arrays.

↳ data stored in tabular form consists of rows & columns.

Doubly linked list :-

↳ complex type of linked list in which a node contains a pointer to the previous as well as the next node in the sequence.

dequeue :-

double ended queue can be defined as an ordered set of elements in which the insertion & deletion can be performed at both the ends, (front & rear)

Quick Sort :-

↳ Sorting algo. ↳ avg O(n log n) ↳ worts [illegible]

↳ utilizes dived & conquer.

↳ divide large array → smaller array.
→ sorting mechanism

Insertion Sort :-

where the sorted array is built having one item at a time

|21 |38 | 29 | 17 | 4 25|

21, 29, 38, 17, 4, 25

(eg. deck of card.)

17, 21, 29, 38, 4, 25

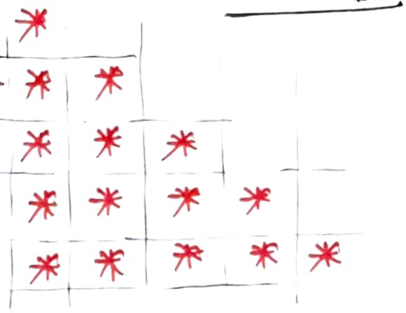
4, 17, 21, 29, 38, 25

4, 17, 21, 25, 29, 38

array elements are compared with each other sequentially & then arrayed.

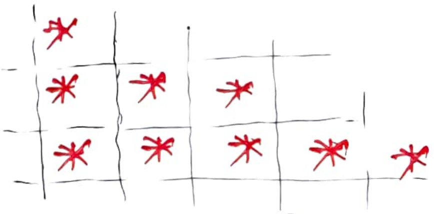
Patterns :-

Ques 1.



```
num= int(input())
for i in range(1, num+1):
    for j in range(1, i+1):
        print("*", end=" ")
    print()
```

Ques.2.

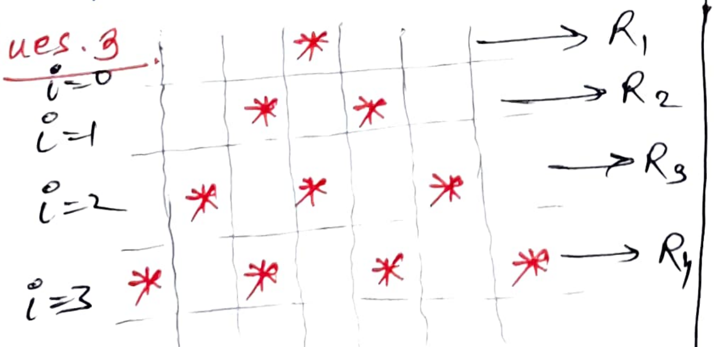


odd no. of stars in the ~~column~~. row.

```
num= int(input( ))
K=1
for i in range(1 , num +1):
    for j in range(1 ,k+1):
        print("*", end=" ")
    K=K+2
    print ()
```

→ rows.
→ coloumn

Ques.3

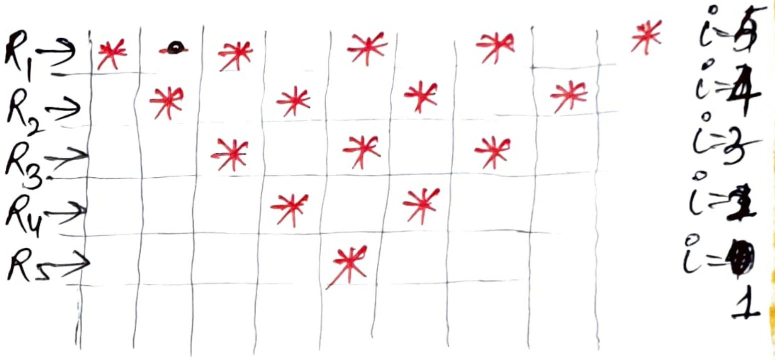


4 rows => num=4
for coloums → space, ·* } 2 for loop

```
num= int(input( ))
for i in range(0, num):
    for j in range(0, num-i-1)
        print(end = " ")
    for j in range(0, i+1):
        print("*", end=" ")
    print()
```

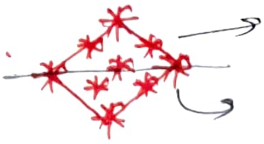
Ques.4.

num=5



```
num =int(input())
for i in range(num, 0, -1):
    for j in range (0, num-i):
        print(end= " ")
    for j in range (0, i) :
        print ("*", end=" ")
    print()
```

Ques. 5.



→ one for loop

→ one for loop

```
def pyramid (rows):
    for i in range (rows):
        print(" "*(rows-i+1)+
                "*"*(i+1))
    for j in range (rows -1, 0, -1):
        print(" "*(rows-j)+
                "*"*(j))
```

DS → spe

Intro

DS → specifies how to organize & manipulate the data.
↳ also tells relationship b/w them.
eg. Linked list, Stack, Queue

Linear DS
↳ if all elements are arranged in sequentiel order.
non-hierarchical

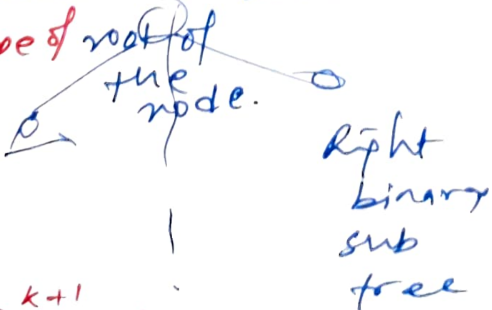
N.L.D.S.
↳ not in seq.

Stack → ordered list (LIFO) ~~insertion~~/del performs at one end.
↳ push & pop., peek
eg. memory managements, fun^c calling & return

Linked list → collection of ~~NL ↳ on bas~~is of data storage randomly stored data obj.
↳ access called nodes.
strategy
- Each node is linked to its adjacent node through a pointer.

Tree → Recursive data structure containing the set of one or more data nodes where one node is designated at the root of the tree while the remaining nodes are called as children.
↳ BT
↳ BST
exp

Binary tree → atmost 2 children
↓
special type of generic tree



$2^{k+1} - 1$

open source lang, easy to debug, Can be integrated with C++/Java
Python: interpreted lang^n.
↳ does not need to compile to run.
↳ well suited to OOP.

general purpose prog. lang

PEP 8 → Py. enhacement proposal

memory managed →
Py. private heap space.
↳ we do not have access to it
↳ py. interpreter takes care of this.
↳ also inbuilt garbage collector.

lambda fun^c → anonymous fun^c. fun^c with no name
add lambda a, b : a+b
add (a, b)
↳ any no. of parameters but just one statement.

dictionary → built in datatype
↳ one to one relationship b/w key & values.

malloc → lib fun^c that allocate memory dynamically.
↳ memory allocated during runtime from the heap segment.

void * malloc (size_t size);

calloc → allocates the memory & also initialize the allocated memory block to zero.

Merge Soot :-

1. Divide the unsorted list into n sublists, each containing one element.

   [list of one 1 element is considered sorted].

2. Repeatedly merge sublists to produce new sorted sublists until there is only one sublist remaining.

time : $O(n \log n)$

space : $O(n)$