Python program:

* Fibonacci series
* Reverse a string
* Factorial of a number
* Palindrome check
* Sorting algorithms (Bubble sort, Insertion sort, Quick sort)
* Prime number check
* Implement a Stack and Queue
* Binary search
* Matrix operations (Addition, Multiplication)
* Find the largest/smallest number in a list.

Python interview question:

* What is Python? What are its advantages?
* What are the different data types in Python?
* How is memory managed in Python?
* What are the differences between a list and a tuple?
* What is the difference between range and xrange in Python?
* How do you handle errors and exceptions in Python?
* What is PEP 8?
* What is the difference between a module and a package?
* What is a lambda function? How do you use it?
* How do you handle file I/O in Python?
* What is a decorator in Python?
* What is a generator in Python? How do you use it?
* What is the difference between deep copy and shallow copy in Python?
* What is Flask?
* What is Django?

Fibonacci series:

Concept: #0 1 1 2 3 5 8 13 21 34

#a b c=a+b

1. n=int(input('Enter the limit: '))
2. a=0
3. if n<=0:
4. print('no negative numbers enter valid number')
5. elif n==1:
6. print(a)
7. **else:**
8. a=0
9. b=1
10. print(a)
11. print(b)
12. for i in range(2,n):
13. c=a+b
14. a=b
15. b=c
16. print(c)

Reverse a string:

1. def reversestring(name):
2. reverse\_string=” ” #empty due to concatenation
3. for i in name: #name='Vicky'
4. reverse\_string=i+reverse\_string
5. #reverse\_string=v+''=v now reverse\_string=v
6. #reverse\_string=i+v=iv now reverse\_string=iv
7. #reverse\_string=c+iv=civ now reverse\_string=civ
8. #reverse\_string=k+civ=kciv now reverse\_string=kciv
9. #reverse\_string=y+civ=ykciv now reverse\_string=ykciv
10. return reverse\_string

Factorial:

Concept: 5! =5\*4\*3\*2\*1

Ex: 5 Ans would be 5\*4\*3\*2\*1=120

1. def factorial(n):
2. factorial=1
3. for i in range(1,n+1):
4. factorial=factorial\*i
5. return factorial

Palindrome:

Concept: Given string should be equal to its reversed string

Ex: madam = madam is a palindrome, Naveen != Naveen is not a palindrome

1. def palindrome(nameorvalue):
2. reverse=nameorvalue[::-1]
3. if nameorvalue==reverse:
4. print(nameorvalue,', is a palindrome')
5. else:
6. print(nameorvalue,', is not a palindrome')

Answers:

1.What is Python? What are its advantages?

Important keywords:

Python, high-level, interpreted, programming language, open-source, easy to learn, simplicity, ease of use, widely used, beginners, large community, support, resources, documentation, versatility, web development, scientific computing, data analysis, artificial intelligence, readability, syntax, maintain, collaborate, codebases, cross-platform, operating systems, Windows, macOS, Linux, standard library, useful modules, functions, integration, multiple languages, productivity.

Python is a high-level, interpreted programming language that was first released in 1991. It is an open-source language that is easy to learn, read and write. Python has gained popularity in recent years because of its simplicity and ease of use, making it one of the most widely used programming languages.

Advantages of Python:

* Easy to Learn: Python is one of the easiest programming languages to learn, with simple and clear syntax that makes it ideal for beginners.
* Large Community and Support: Python has a huge community of developers, which means that there are numerous resources and documentation available to help you solve problems and learn new things.
* Versatility: Python can be used for a variety of tasks such as web development, scientific computing, data analysis, artificial intelligence, and more.
* Readability: Python's syntax is easy to read and understand, making it easier to maintain and collaborate on large codebases.
* Cross-platform: Python code can run on different operating systems, including Windows, macOS, and Linux.
* Standard Library: Python comes with a rich standard library that includes many useful modules and functions, making it easy to perform various tasks.
* Integration: Python can easily be integrated with other programming languages, allowing you to combine the strengths of multiple languages in a single project.
* Productivity: Python's simplicity and ease of use allow developers to write code quickly and efficiently, saving time and increasing productivity.