Q.1 why high level language-

**Python is easy to use, Python runs on any platform, Extensive support libraries, Python is accessible, Incredible Artificial Intelligence and Machine Learning support**

**General purpose language??- a programming language for building software in a wide variety of application domains**

**Q.2**

**in Python, the type of a variable is determined at runtime rather than at compile time**.

Q.3

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| Large Community | Slower than compiled languages |
| Flexible and Extensible | Security |
| Extensive Libraries | Work Environment |
| Embeddable | High memory consumption |

Q.4

**artificial intelligence, machine learning and deep learning**

**Q.5**

**Python** has no command for **declaring** a **variable**. A **variable** is created when some value is assigned **to** it.

e.g

name=”prakhar” here name is variable and prakhar is value assigned to name variable.

Q.6

Using input() function

Q.7

String

Q.8

Conversion of one datatype to another data type

Q.9

Yes

x, y = input("Enter First Name: "), input("Enter Last Name: ")

Q.10

Python keywords are **special reserved words that have specific meanings and purposes and can't be used for anything but those specific purposes**.

Q.11

No coz we cannot use keywords for any other purpose except for special purspose that for keywords are.

Q.12

Python indentation is a way **of telling a Python interpreter that the group of statements belongs to a particular block of code**.

Q.13

Using print() function

Q.14

That performs operations on operands

Q.15

/ - is use for float division

//-is use for integer division

Q.16

Str=”iNeuron”\*4

Print(str)

Q.17

num=int(input("Enter the number: "))

if num%2==0:

    print("even")

else:

    print("odd")

Q.18

Boolean Operators are **simple words (AND, OR, NOT or AND NOT) used as conjunctions to combine or exclude keywords in a search**

**Q.19**

**1**

**0**

**False**

**1**

**Q.20**

**used to handle conditions in your program**.

Q.21

if… elif…else are conditional statements that **provide you with the decision making that is required when you want to execute code based on a particular condition**.

Q.22

age=int(input("Enter the age: "))

if age>=18:

    print("i can vote")

else:

    print("i can't vote2")

**Q.23**

numbers = [12, 75, 150, 180, 145, 525, 50]

sum=0

for i in numbers:

    sum=sum+i

print(sum)

**Q.24**

num1=int(input("Enter 1st number: "))

num2=int(input("Enter 2nd number: "))

num3=int(input("Enter 3rd number: "))

if num1>num2 and num1>num3:

    print(num1)

elif num2>num1 and num2>num3:

    print(num2)

else:

    print(num3)

**Q.25**

numbers = [12, 75, 150, 180, 145, 525, 50]

for num in numbers:

    if num%5==0:

        print(num)

    if num>150:

        continue

    if num>500:

        print(num)

        break

Q.26

String is a sequence of character .

Syntax=>

String1=”Hello My name is Prakhar Rai”

QA.27

String is has index starting from 0 so we can access it via their index.

E.gstring1="hello Prakhar"

#want to acces the fist letter of the string that is h so??

print(string1[0])

Q.28

string = "Big Data iNeuron"

desired\_output = "iNeuron"

string = "Big Data iNeuron"

print(string[9:])

Q.29

string = "Big Data iNeuron"

desired\_output = "norueNi"

string = "Big Data iNeuron"

print(string[-1:-8:-1])

Q.29

string = "Big Data iNeuron"

rev\_string=string[-1:-8:-1]

print(rev\_string)

Q.30

string = "Big Data iNeuron"

rev\_string=string[-1::-1]

print(rev\_string)

Q.31

string = "Big Data iNeuron"

print(string)

del(string)

print(string)

Q.32

An escape sequence is **a sequence of characters that, when used inside a character or string, does not represent itself but is converted into another character or series of characters**. Eg. \n –new line,\’-single quote

Q.33

string = "'iNeuron's Big Data Course'"

print(string)

Q.34

It stores hetrogenous kind of data,it stores sequential data in contagious memory location

Q.35

List1=[] empty list

List2=[1,2,”hello”,23]

Q.36

List2=[1,2,"hello",23]

#first element access

print(List2[0])

# we access list element via index starting form 0

Q.37

lst = [1,2,3,"Hi",[45,54, "iNeuron"], "Big Data"]

print(lst[4][2])

Q.38

list1=[]

n=int(input("Enter how many values you want to enter in list: "))

for i in range(0,n):

    ele=int(input())

    list1.append(ele)

print(len(list1))

Q.39

lst = ["Welcome", "to", "Data", "course"]

lst.insert(3, "Big")

print(lst)

Q.40

Tuples also stores hetrogenous sequence of data but it is immutable unlike list and like string. All the functionalities works for tuple like list such as indexing,iteration but we cannot perform append,delete,updation in tuple like list.

Q.41

T1=() #empty tuple

T2=(20,30,40,50,60)

Q.42

t1=(1,"hello","how",2)

print(t1)

#we cannot add my name in tuple coz tuple is immutable and we cannot perfirm append command.

Q.43

t1=(1,"hello","how",2)

t2=(1,2,3)

print(t1+t2)

#we cannot append two tuple but we can concatenate two tuple using + operator that will create a new tuple and will not appedn the first tuple into the second tuple.

Q.44

list=[]

t1=()

n=int(input("enter the number of element you wnat to eneter in tuple: "))

count=0

for i in range(0,n):

    ele=input()

    count+=1

    list.append(ele)

t1=tuple(list)

print(count)

Q.45

Set is a data structure in python that retruns distinct values in arranged order.

Q.46

Set1=set() #empty set

Set2={1,2,3,4,5,6}

Q.47

set1=set()

set1.add("iNeuron")

print(set1)

Q.48

set1=set()

set1.add("iNeuron")

print(set1)

set1.add(1)

set1.add(2)

set1.add(3)

set1.add(4)

set1.add(5)

print(set1)

Q.49

set1=set()

set1.add("iNeuron")

print(set1)

set1.add(1)

set1.add(2)

set1.add(3)

set1.add(4)

set1.add(5)

print(set1)

set2={"hello",1,2,"add me in set1"}

set1.update(set2)

print(set1)

#update function can be used to add another set to the current set where add function add values to the current set

Q.50

set1=set()

set1.add("iNeuron")

print(set1)

set1.clear()

print(set1)

#clear () removes all the values from the set

Q.51

Frozenset is similar to set in Python, except that **frozensets are immutable**, which implies that once generated, elements from the frozenset cannot be added or removed. This function accepts any iterable object as input and transforms it into an immutable object.

Q.52

Frozenset is similar to set in Python, except that **frozensets are immutable**, which implies that once generated, elements from the frozenset cannot be added or removed. This function accepts any iterable object as input and transforms it into an immutable object.

Q.53

set1=set()

set1.add("iNeuron")

set1.add(1)

set1.add(2)

set1.add(3)

set2={1,2,3,4}

print(set1|set2)

#union in sets perform by '|'  that is totally means it takes all the  values from both the sets

Q.54

set1=set()

set1.add("iNeuron")

set1.add(1)

set1.add(2)

set1.add(3)

set2={1,2,3,4}

print(set1&set2)

#union in sets perform by '&'  that is totally means it takes all the common values from both the sets

Q.55

Dictionary is data structure in python use to store values in key values structure.

Q.56

The dictionary Data Structure in Python is an unordered collection of items. While other Data Structures use only one value as the element, **the dictionary is a slightly more compound data structure**. It makes use of two elements i.e. a pair of elements, namely, a key and a value

Q.57

Dict1={}#empty dictionary

Dict2={"name":"prakhar","age":22,"hello":[1,2,3,4,5],"other\_details":"city\_gorakhpur"}

print(Dict2)

Q.58

<class 'dict'>

Q.59

Dict1={}#empty dictionary

Dict2={"name":"prakhar","age":22,"hello":[1,2,3,4,5],"other\_details":"city\_gorakhpur"}

print(Dict2)

Dict1["name"]="prakhar"

Dict1["age"]=22

Dict1["location"]="Gorakhpur"

print(Dict1)

Q.60

Dict1={}#empty dictionary

Dict2={"name":"prakhar","age":22,"hello":[1,2,3,4,5],"other\_details":"city\_gorakhpur"}

print(Dict2)

Dict1["name"]="prakhar"

Dict1["age"]=22

Dict1["location"]="Gorakhpur"

print(Dict1)

Dict1\_values=list(Dict1.values())

print(Dict1\_values)

Q.61

dict1={"name":'prakhar',"age":22,"other\_details":{"loaction":"gorakhapur","village":"tiha"}}

print(dict1["other\_details"])

Q.62

dict1={"name":'prakhar',"age":22,"other\_details":{"loaction":"gorakhapur","village":"tiha"}}

print(dict1.get("name"))

#it retrun the values for the specific key.

Q.63

dict1={"name":'prakhar',"age":22,"other\_details":{"loaction":"gorakhapur","village":"tiha"}}

for k,v in dict1.items():

    print("Key is: ",k,"value is :",v)

#it retrun all the keys values for teh dictionary

Q.64

dict1={"name":'prakhar',"age":22,"other\_details":{"loaction":"gorakhapur","village":"tiha"}}

dict1.pop("name")

print(dict1)

#it removes the values form the dictionary from the specific key and in ,list removes values form the specific index

Q.65

car={"name":"BMW","Model":2019,"color":"black"}

print(car)

car.popitem()

print(car)

#it removes the last item inserted into the dictionary

Q.66

car={"name":"BMW","Model":2019,"color":"black"}

keys\_dict1=list(car.keys())

print(keys\_dict1)

#it returns all the keys from dictionary

Q.67

car={"name":"BMW","Model":2019,"color":"black"}

keys\_dict1=list(car.values())

print(keys\_dict1)

#it returns all the values from dictionary

Q.68

Loops is used to iterate through sequence.

Q.69

Three types- For loop,while loop,nested loop(loop inside loop)

Q.70

For loop is used when number of iteration is known

While Loop is used when number of iteration is not known.

Q.71The continue keyword is used **to end the current iteration in a for loop (or a while loop), and continues to the next iteration**.

Q.72

Break' in Python is a loop control statement. It is used **to control the sequence of the loop**. Suppose you want to terminate a loop and skip to the next code after the loop; break will help you do that.

Q.73

pass **tells Python to skip this line and do nothing**.

Q.74

The range() function **returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and stops before a specified number**.

Q.75

Yes we can loop over a dictionary.

Q.76

N=int(input("Enter the number for the factorial: "))

fact=1

while(N>0):

    fact=fact\*N

    N=N-1

print(fact)

Q.77

P=int(input())

R=int(input())

T=int(input())

SI=(P\*R\*T)/100

print(SI)

Q.78

P=int(input())

R=int(input())

T=int(input())

A=P\*(1+R/100)\*\*T

print(A)

Q.79

n=int(input("Enter a number to check prime or not: "))

flag=False

if(n==1):

    flag=True

if(n>1):

    for i in range(2,n):

        if(n%i)==0:

            flag=True

            break

if(flag==True):

    print("not prime")

else:

    print("prime")

Q.80n=int(input("Enter a number to check armstrong or not: "))

p=n

sum=0

while(p>0):

    r=p%10

    sum=sum+r\*r\*r

    p=p//10

if(n==sum):

    print("Armstrong")

else:

    print("not Armstrong")

Q.81

Q.82

list1=[1,2,3,4,5,6,7,8,9]

a=list1[len(list1)-1]

b=list1[0]

list1[0]=a

list1[len(list1)-1]=b

print(list1)

Q.83

list1=[1,2,3,4,5,6,7,8,9]

a=list1[len(list1)-1]

b=list1[0]

list1[0]=a

list1[len(list1)-1]=b

print(list1)

Q.84

list1=[2,1,9,4,7,3,8,6]

list1.sort()

print(list1[len(list1)-1])

Q.85

list1=[2,1,9,4,7,3,8,6]

sum=0

for i in list1:

    sum=sum+i

print(sum)

Q.86

# str=input("Enter the string to check the palindrome: ")

str1=input("Enter the String to check palindrome or not: ")

str2=str1[-1::-1]

if(str2==str1):

    print("palindrome")

else:

    print("not palindrome")

Q,87

i=int(input("Enter the ith number to remove from string: "))

string="hello i love you"

a=string[i-1]

new\_string=string.replace(a,"")

print(new\_string)

Q.88

Q.89

Q.90