```
{
 "cells": [
  {
   "cell_type": "markdown",
   "id": "96db7429-3919-467b-b9cc-47594fcd2639",
   "metadata": {},
   "source": [
    "#Question(1)_Discuss string slicing and provide examples."
   ]
  },
  {
   "cell_type": "markdown",
   "id": "4d4ef204-aeb9-4515-92dc-a6e73c7f18b1",
   "metadata": {},
   "source": [
    "##Answer_'''String slicing in python allows extract a portion of a string by specifying a start and end index.\n",
    "           It follows the format string[start:end],wherestart is the index where slicing begins and end is the index where it ends.'''"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 4,
   "id": "f503f1f9-6bc4-4021-a9c8-481e2821a1fd",
   "metadata": {},
   "outputs": [],
   "source": [
    "string = \"My name is Manisha\""
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 10,
   "id": "40c6d97f-e552-461e-b8f3-cd49cfbba89b",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "'My na'"
      ]
     },
     "execution_count": 10,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "string [ 0 : 5 ]        # 5 is exclusive, it will give result before 5"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 12,
   "id": "6d4270d1-af28-48a8-9cd0-99c2076fc6c0",
   "metadata": {},
   "outputs": [
```

```
    {
     "data": {
      "text/plain": [
       "' Mani'"
      ]
     },
     "execution_count": 12,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "string [10 : 15 ]"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 14,
   "id": "1da49954-7595-4cf6-b442-2cd4cde325f8",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "'My name is Manisha'"
      ]
     },
     "execution_count": 14,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "string [ 0 : ]          #start_index:end_index  >> if you not provide end idex, it will be default give substring till the last index"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 16,
   "id": "682d36e1-a9c4-4f0a-920e-9d30ab18d217",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "'me is Manisha'"
      ]
     },
     "execution_count": 16,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "string [ 5 : ]"
   ]
  },
  {
```

```
{
 "cell_type": "code",
 "execution_count": 23,
 "id": "db90ea5b-6d50-45ef-b9fe-f5c3c0d7d03f",
 "metadata": {},
 "outputs": [],
 "source": [
  "string1 = \"success\""
 ]
},
{
 "cell_type": "code",
 "execution_count": 25,
 "id": "e2e4a41a-c4fd-49e5-9c08-05a5e3420ab8",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "'s'"
    ]
   },
   "execution_count": 25,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "string1 [-1]"
 ]
},
{
 "cell_type": "code",
 "execution_count": 29,
 "id": "7f4ab5ed-4ca1-407c-a68f-c17efd754865",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "'succes'"
    ]
   },
   "execution_count": 29,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "string1 [ : -1]                    # except the last one character"
 ]
},
{
 "cell_type": "code",
 "execution_count": 31,
 "id": "fa605f32-5a2d-44d1-b188-8a0d9bc05c2c",
 "metadata": {},
 "outputs": [
```

```
  {
   "data": {
    "text/plain": [
     "'ccess'"
    ]
   },
   "execution_count": 31,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "string1 [-5 : ]                    # only get last five characters"
 ]
},
{
 "cell_type": "code",
 "execution_count": 33,
 "id": "143575fb-3882-427d-bf52-844e062b68eb",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "'succe'"
    ]
   },
   "execution_count": 33,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "string1 [ 0 : 5 : 1 ]                # by default step is 1 "
 ]
},
{
 "cell_type": "code",
 "execution_count": 35,
 "id": "f5d4b09a-9ed1-4141-a1c8-8e8e06a78113",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "'sce'"
    ]
   },
   "execution_count": 35,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "string1 [ 0 : 5 : 2 ]                # syntax [start:end:step]"
 ]
},
{
```

```
  "cell_type": "markdown",
  "id": "d7faa9d3-92ac-4155-90b2-352d43917695",
  "metadata": {},
  "source": [
   "#Question(2)_Explain the key features of lists in Python."
  ]
 },
 {
  "cell_type": "markdown",
  "id": "5dbf8933-aa3c-47b1-b816-da770d5b5c57",
  "metadata": {},
  "source": [
   "##Answer_'''List is an ordered collection of elements that can be of any data type.\n",
   "          Lists can hold/store hetrogenous data (numbers, strings, even other lists).\n",
   "          List are mutable,You can add, remove, or modify elements within a list using indexing and slicing. \n",
   "          Lists are versatile for storing and managing collections that might change.\n",
   "          List are ordered,newly added items will be placed at the end of the list.\n",
   "          List use zero-based indexing,every list item has an associated index,and the list item's index is 0."""
  ]
 },
 {
  "cell_type": "markdown",
  "id": "fa7008b5-1542-488e-8744-3cf5ae9a2b6a",
  "metadata": {},
  "source": [
   "#Question(3)_Describe how to access, modify, and delete elements in a list with examples."
  ]
 },
 {
  "cell_type": "markdown",
  "id": "c2d1a45d-c939-42cb-978f-18ac81da6846",
  "metadata": {},
  "source": [
   "##Answer_List are mutable,it can store any datatype( heterogenous data)."
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 54,
  "id": "334fac29-b0a6-4e54-961b-b33e9fd36e6f",
  "metadata": {},
  "outputs": [],
  "source": [
   "A = [1,2,3,1.2,3+4j,\"milk\",\"potato\",\"pasta\",True,False,1,1,2,5]"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 56,
  "id": "52b6ee17-ee10-4e4a-aec6-588a990f01c6",
  "metadata": {},
  "outputs": [
   {
    "data": {
     "text/plain": [
      "list"
```

```
    ]
   },
   "execution_count": 56,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "type(A)"
 ]
},
{
 "cell_type": "code",
 "execution_count": 58,
 "id": "117d7c30-24ad-437c-b0cc-d97818ede483",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "1"
    ]
   },
   "execution_count": 58,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "A[0]"
 ]
},
{
 "cell_type": "code",
 "execution_count": 60,
 "id": "8d5d9bcb-004d-468e-a07d-473f702f34ae",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "'milk'"
    ]
   },
   "execution_count": 60,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "A[5]"
 ]
},
{
 "cell_type": "code",
 "execution_count": 62,
 "id": "73612552-e6bd-4e55-b604-3e4e7337a67a",
 "metadata": {},
```

```
"outputs": [],
"source": [
 "#adding element to the list\n",
 "A.append(\"mobile\")"
]
},
{
"cell_type": "code",
"execution_count": 64,
"id": "357605d0-1fec-4ea5-b7e2-c6c9df4d984b",
"metadata": {},
"outputs": [
 {
  "data": {
   "text/plain": [
    "[1,\n",
    " 2,\n",
    " 3,\n",
    " 1.2,\n",
    " (3+4j),\n",
    " 'milk',\n",
    " 'potato',\n",
    " 'pasta',\n",
    " True,\n",
    " False,\n",
    " 1,\n",
    " 1,\n",
    " 2,\n",
    " 5,\n",
    " 'mobile']"
   ]
  },
  "execution_count": 64,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "A"
]
},
{
"cell_type": "code",
"execution_count": 66,
"id": "aeab59c2-2d57-40e8-a5ff-903234886bad",
"metadata": {},
"outputs": [],
"source": [
 "#remove element to the list\n",
 "A.remove(\"milk\")"
]
},
{
"cell_type": "code",
"execution_count": 68,
"id": "eb39d2c4-9642-46a5-98fe-137c0516c996",
"metadata": {},
"outputs": [
```

```
 {
  "data": {
   "text/plain": [
    "[1, 2, 3, 1.2, (3+4j), 'potato', 'pasta', True, False, 1, 1, 2, 5, 'mobile']"
   ]
  },
  "execution_count": 68,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "A"
]
},
{
"cell_type": "code",
"execution_count": 70,
"id": "c33203a0-c342-4ad4-a543-44bacb208826",
"metadata": {},
"outputs": [
 {
  "data": {
   "text/plain": [
    "'potato'"
   ]
  },
  "execution_count": 70,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "A[5]"
]
},
{
"cell_type": "code",
"execution_count": 76,
"id": "1b91d999-712c-4430-8594-093981029514",
"metadata": {},
"outputs": [],
"source": [
 "A[5] = \"pen\""
]
},
{
"cell_type": "code",
"execution_count": 78,
"id": "66c7286e-3240-43c2-a53e-1a6df902464c",
"metadata": {},
"outputs": [
 {
  "data": {
   "text/plain": [
    "[1, 2, 3, 1.2, (3+4j), 'pen', 'pasta', True, False, 1, 1, 2, 5, 'mobile']"
   ]
  },
```

```
  "execution_count": 78,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "A"
]
},
{
"cell_type": "code",
"execution_count": 92,
"id": "2e9637f9-834c-41e0-af62-c46e709b1d81",
"metadata": {},
"outputs": [],
"source": [
 "del A"
]
},
{
"cell_type": "code",
"execution_count": 94,
"id": "079aa63d-7d1d-4315-b61f-c386b6889eee",
"metadata": {},
"outputs": [
 {
  "ename": "NameError",
  "evalue": "name 'A' is not defined",
  "output_type": "error",
  "traceback": [
   "\u001b[1;31m---------------------------------------------------------------------------\u001b[0m",
   "\u001b[1;31mNameError\u001b[0m                                 Traceback (most recent call last)",
   "Cell \u001b[1;32mIn[94], line 1\u001b[0m\n\u001b[1;32m----> 1\u001b[0m A\n",
   "\u001b[1;31mNameError\u001b[0m: name 'A' is not defined"
  ]
 }
],
"source": [
 "A                                    # It deletes the variable list."
]
},
{
"cell_type": "code",
"execution_count": 84,
"id": "7ab89be1-55d3-490f-a8a0-965ba2224bf6",
"metadata": {},
"outputs": [],
"source": [
 "A = [1,2,3,1.2,3+4j,\"milk\",\"potato\",\"pasta\",True,False,1,1,2,5]"
]
},
{
"cell_type": "code",
"execution_count": 86,
"id": "8bd86ff7-e160-444d-be65-61e642289eb1",
"metadata": {},
"outputs": [],
"source": [
```

```
 "A.clear()                    # It deletes the element of list,but variable list will not be deleted."
 ]
},
{
 "cell_type": "code",
 "execution_count": 88,
 "id": "51e34d25-9c72-4db5-a488-99137e82d3ae",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "[]"
    ]
   },
   "execution_count": 88,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "A"
 ]
},
{
 "cell_type": "markdown",
 "id": "8a495f0b-fccd-4f4d-983c-1c6b0e6753e3",
 "metadata": {},
 "source": [
  "#Question(4)_ Compare and contrast tuples and lists with examples."
 ]
},
{
 "cell_type": "markdown",
 "id": "24539a87-42c1-4f48-b96f-15e81eaf5694",
 "metadata": {},
 "source": [
  "##Answer_Tuples and list are ordered collection of elements(heterogenous data stored)."
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
 "id": "f1c8de47-da47-4faf-8400-eb1eecdb8ae9",
 "metadata": {},
 "outputs": [],
 "source": [
  "# Tuples are immutable\n",
  "# t ()"
 ]
},
{
 "cell_type": "code",
 "execution_count": 100,
 "id": "026a50c7-1bbd-4e18-8fde-14628e0a60a7",
 "metadata": {},
 "outputs": [],
 "source": [
```

```
  "t = (1,2,3,1.2,3+4j,\"milk\",\"potato\",\"pasta\",True,False,1,1,2,5)"
 ]
},
{
 "cell_type": "code",
 "execution_count": 102,
 "id": "da0cce0e-9354-438f-a01e-4cc5c9b0b445",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "(1, 2, 3, 1.2, (3+4j), 'milk', 'potato', 'pasta', True, False, 1, 1, 2, 5)"
    ]
   },
   "execution_count": 102,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "t"
 ]
},
{
 "cell_type": "code",
 "execution_count": 104,
 "id": "0a7d162a-9c00-4c17-9a63-c697e80cbe0f",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "tuple"
    ]
   },
   "execution_count": 104,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "type (t)"
 ]
},
{
 "cell_type": "code",
 "execution_count": 108,
 "id": "2711e2e6-e3ad-4c0b-8543-db2c40e4bc45",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "2"
    ]
   },
   "execution_count": 108,
```

```
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "t [1]"
 ]
},
{
 "cell_type": "code",
 "execution_count": 140,
 "id": "72b0c9c6-7ffb-4cd1-bc6b-a4be3e826d58",
 "metadata": {},
 "outputs": [
  {
   "ename": "TypeError",
   "evalue": "'tuple' object does not support item assignment",
   "output_type": "error",
   "traceback": [
    "\u001b[1;31m---------------------------------------------------------------------------\u001b[0m",
    "\u001b[1;31mTypeError\u001b[0m                                 Traceback (most recent call last)",
    "Cell \u001b[1;32mIn[140], line 1\u001b[0m\n\u001b[1;32m----> 1\u001b[0m t
[\u001b[38;5;241m1\u001b[39m] \u001b[38;5;241m=\u001b[39m \u001b[38;5;241m5\u001b[39m\n",
    "\u001b[1;31mTypeError\u001b[0m: 'tuple' object does not support item assignment"
   ]
  }
 ],
 "source": [
  "t [1] = 5                    # tuple as a data structure where dont want to modify the data\n",
  "                            # use tuple to store the data.(example,adhar card, ATM no.,employ id etc.)"
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
 "id": "2d4811ca-fa85-4894-b5d9-3c7136dbe728",
 "metadata": {},
 "outputs": [],
 "source": [
  "# List is mutable"
 ]
},
{
 "cell_type": "code",
 "execution_count": 126,
 "id": "87f3d3b9-c51a-41b3-9f85-e3c411b93c9c",
 "metadata": {},
 "outputs": [],
 "source": [
  "l = [1,2,3,1.2,3+4j,\"milk\",\"potato\",\"pasta\",True,False,1,1,2,5]"
 ]
},
{
 "cell_type": "code",
 "execution_count": 128,
 "id": "503ae364-5572-4ab0-b0a9-0cd93d7535ee",
 "metadata": {},
 "outputs": [
```

```
  {
   "data": {
    "text/plain": [
     "[1, 2, 3, 1.2, (3+4j), 'milk', 'potato', 'pasta', True, False, 1, 1, 2, 5]"
    ]
   },
   "execution_count": 128,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "l"
 ]
},
{
 "cell_type": "code",
 "execution_count": 130,
 "id": "46750a2d-4675-486a-8514-738c6d393e8d",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "list"
    ]
   },
   "execution_count": 130,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "type (l)"
 ]
},
{
 "cell_type": "code",
 "execution_count": 118,
 "id": "8be677c3-5665-4a65-9e82-374c6f95efe8",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "2"
    ]
   },
   "execution_count": 118,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "l [1]"
 ]
},
{
```

```
  "cell_type": "code",
  "execution_count": 134,
  "id": "0c65678c-8794-4f93-8437-017176249438",
  "metadata": {},
  "outputs": [],
  "source": [
   "l [1] = 5"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 138,
  "id": "6a7248db-b833-4485-bbb4-4fe19cf4c50e",
  "metadata": {},
  "outputs": [
   {
    "data": {
     "text/plain": [
      "[1, 5, 3, 1.2, (3+4j), 'milk', 'potato', 'pasta', True, False, 1, 1, 2, 5]"
     ]
    },
    "execution_count": 138,
    "metadata": {},
    "output_type": "execute_result"
   }
  ],
  "source": [
   "l                                  # List as a data structure where to modify the data."
  ]
 },
 {
  "cell_type": "markdown",
  "id": "88ce31b4-81d6-4922-b89f-e0c038cfe263",
  "metadata": {},
  "source": [
   "#Question(5)_ Describe the key features of sets and provide examples of their use."
  ]
 },
 {
  "cell_type": "markdown",
  "id": "32bbf3b2-e6a2-4ca8-8d95-ad09b14ceaa7",
  "metadata": {},
  "source": [
   "##Answer_'"sets are Unordered(indexing will not work) collections of unique elements. The order doesn't matter, and duplicate entries are not allowed. \n",
   "            Sets are useful for checking membership (if an item exists) or finding the intersection/difference between sets.\n",
   "             Set operations like union (combining elements), intersection (finding common elements), and difference (finding elements in one set but no                the other) are efficient'"."
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 144,
  "id": "b40102fb-6059-417f-b122-00a5ec2c858c",
  "metadata": {},
  "outputs": [],
  "source": [
```

```
 "s = {}"
]
},
{
"cell_type": "code",
"execution_count": 166,
"id": "cab5e6e2-9884-4ba9-bf4c-1748bf6883cf",
"metadata": {},
"outputs": [],
"source": [
 "s = {1,2,3,1.2,3+4j,\"milk\",\"potato\",\"pasta\",True,False,1,1,2,5}"
]
},
{
"cell_type": "code",
"execution_count": 168,
"id": "ef4e7a38-b8d0-4d0b-9396-43a56a9d1236",
"metadata": {},
"outputs": [
 {
  "data": {
   "text/plain": [
    "{(3+4j), 1, 1.2, 2, 3, 5, False, 'milk', 'pasta', 'potato'}"
   ]
  },
  "execution_count": 168,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "s                                    # Does not allow duplicate element."
]
},
{
"cell_type": "code",
"execution_count": 174,
"id": "96c2c9c9-217c-4aee-9c4f-6a6ad6079388",
"metadata": {},
"outputs": [],
"source": [
 "s.add(\"apple\")"
]
},
{
"cell_type": "code",
"execution_count": 176,
"id": "f7c52846-cb28-48e0-a8c6-8ce9664e0c8f",
"metadata": {},
"outputs": [
 {
  "data": {
   "text/plain": [
    "{(3+4j), 1, 1.2, 2, 3, 5, False, 'apple', 'milk', 'pasta', 'potato'}"
   ]
  },
  "execution_count": 176,
  "metadata": {},
```

```
    "output_type": "execute_result"
   }
  ],
  "source": [
   "s"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 178,
  "id": "9c5ffd30-697f-4d18-91ab-73b9329a9a97",
  "metadata": {},
  "outputs": [],
  "source": [
   "s.remove(2)"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 180,
  "id": "cbd12536-0626-4fe6-a96a-3d887fc4834b",
  "metadata": {},
  "outputs": [
   {
    "data": {
     "text/plain": [
      "{(3+4j), 1, 1.2, 3, 5, False, 'apple', 'milk', 'pasta', 'potato'}"
     ]
    },
    "execution_count": 180,
    "metadata": {},
    "output_type": "execute_result"
   }
  ],
  "source": [
   "s"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 182,
  "id": "0576316c-1044-472d-9cc9-08a8a5107b28",
  "metadata": {},
  "outputs": [
   {
    "ename": "TypeError",
    "evalue": "'set' object is not subscriptable",
    "output_type": "error",
    "traceback": [
     "\u001b[1;31m---------------------------------------------------------------------------\u001b[0m",
     "\u001b[1;31mTypeError\u001b[0m                                 Traceback (most recent call last)",
     "Cell \u001b[1;32mIn[182], line 1\u001b[0m\n\u001b[1;32m----> 1\u001b[0m s[\u001b[38;5;241m1\u001b[39m]\n",
     "\u001b[1;31mTypeError\u001b[0m: 'set' object is not subscriptable"
    ]
   }
  ],
  "source": [
```

```
   "s[1]                                  # indexing will not allowed.\n",
   "                                      # add or remove elements from a set, but cannot access elements by index
(since order doesn't matter)"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 160,
  "id": "91a2f70a-c8d6-4750-b5de-2eef6bb9afcc",
  "metadata": {},
  "outputs": [],
  "source": [
   "s = {1,2,3,4,5,6}"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 162,
  "id": "4e78170d-fca2-41ef-ba93-a7e1dfd6afff",
  "metadata": {},
  "outputs": [
   {
    "data": {
     "text/plain": [
      "{1, 2, 3, 4, 5, 6}"
     ]
    },
    "execution_count": 162,
    "metadata": {},
    "output_type": "execute_result"
   }
  ],
  "source": [
   "s"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 184,
  "id": "ede61fa6-646e-4bf5-922a-32714db31775",
  "metadata": {},
  "outputs": [
   {
    "ename": "TypeError",
    "evalue": "unhashable type: 'set'",
    "output_type": "error",
    "traceback": [
     "\u001b[1;31m---------------------------------------------------------------------------\u001b[0m",
     "\u001b[1;31mTypeError\u001b[0m                                 Traceback (most recent call last)",
     "Cell \u001b[1;32mIn[184], line 1\u001b[0m\n\u001b[1;32m----> 1\u001b[0m s
\u001b[38;5;241m=\u001b[39m
{\u001b[38;5;241m1\u001b[39m,\u001b[38;5;241m2\u001b[39m,\u001b[38;5;241m3\u001b[39m,\u001b[38;5;241m4\u001b[39m,\u001b[38;5;241m5\u001b[39m,{\u001b[38;5;241m5\u001b[39m,\u001b[38;5;241m6\u001b[39m,\u001b[38;5;241m7\u001b[39m}}\n",
     "\u001b[1;31mTypeError\u001b[0m: unhashable type: 'set'"
    ]
   }
  ],
```

```
  "source": [
   "s = {1,2,3,4,5,{5,6,7}}"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 186,
  "id": "539ad647-a451-44db-9056-fe5e6075fedd",
  "metadata": {},
  "outputs": [
   {
    "ename": "TypeError",
    "evalue": "unhashable type: 'list'",
    "output_type": "error",
    "traceback": [
     "\u001b[1;31m---------------------------------------------------------------------------\u001b[0m",
     "\u001b[1;31mTypeError\u001b[0m                                 Traceback (most recent call last)",
     "Cell \u001b[1;32mIn[186], line 1\u001b[0m\n\u001b[1;32m----> 1\u001b[0m s \
\u001b[38;5;241m=\u001b[39m \
{\u001b[38;5;241m1\u001b[39m,\u001b[38;5;241m2\u001b[39m,\u001b[38;5;241m3\u001b[39m,\u001b[38;5;241m4\u001b[39m,\u001b[38;5;241m5\u001b[39m,[\u001b[38;5;241m5\u001b[39m,\u001b[38;5;241m6\u001b[39m,\u001b[38;5;241m7\u001b[39m]}\n",
     "\u001b[1;31mTypeError\u001b[0m: unhashable type: 'list'"
    ]
   }
  ],
  "source": [
   "s = {1,2,3,4,5,[5,6,7]}                    # unhashable type >> list and set is mutable"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 188,
  "id": "6ba10afc-64cd-4646-a011-158ca26d330c",
  "metadata": {},
  "outputs": [],
  "source": [
   "s = {1,2,3,4,5,(5,6,7)}"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 192,
  "id": "d2e8a72e-4cab-41f4-807d-03e77df861f3",
  "metadata": {},
  "outputs": [
   {
    "data": {
     "text/plain": [
      "{(5, 6, 7), 1, 2, 3, 4, 5}"
     ]
    },
    "execution_count": 192,
    "metadata": {},
    "output_type": "execute_result"
   }
  ],
  "source": [
```

```
 "s                                  # tuple is immutable data structure has a stable hash values."
 ]
},
{
 "cell_type": "code",
 "execution_count": 194,
 "id": "fe7831ea-5ff7-410a-986f-43c6372787b2",
 "metadata": {},
 "outputs": [],
 "source": [
  "# set operations\n",
  "# Union : combines elements from two sets excluding duplicates"
 ]
},
{
 "cell_type": "code",
 "execution_count": 196,
 "id": "b795eb03-6415-4180-8f26-b860a9a22f8e",
 "metadata": {},
 "outputs": [],
 "source": [
  "s1 = {\"apple\",\"banana\",\"orange\",\"milk\",\"pen\"}"
 ]
},
{
 "cell_type": "code",
 "execution_count": 198,
 "id": "e176c33e-b004-4806-9a68-cb8049e5b712",
 "metadata": {},
 "outputs": [],
 "source": [
  "s2 = {\"mobile\",\"bottle\",\"orange\",}"
 ]
},
{
 "cell_type": "code",
 "execution_count": 200,
 "id": "48a9fe59-fdc0-4688-bebf-bc3d2628b0b6",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "{'apple', 'banana', 'bottle', 'milk', 'mobile', 'orange', 'pen'}"
    ]
   },
   "execution_count": 200,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "s1 | s2"
 ]
},
{
 "cell_type": "code",
 "execution_count": 202,
```

```
 "id": "b53f667b-f512-471d-870b-2e38afeb9302",
 "metadata": {},
 "outputs": [],
 "source": [
  "# Intersection : Only common elements between sets."
 ]
},
{
 "cell_type": "code",
 "execution_count": 206,
 "id": "4ab675ae-bba0-434b-b743-093c433796a7",
 "metadata": {},
 "outputs": [],
 "source": [
  "s1 = {\"apple\",\"banana\",\"orange\",\"milk\",\"pen\"}"
 ]
},
{
 "cell_type": "code",
 "execution_count": 208,
 "id": "7f43cdc7-7334-4fb9-90b9-b4f6bebd8bf0",
 "metadata": {},
 "outputs": [],
 "source": [
  "s2 = {\"mobile\",\"bottle\",\"orange\",}"
 ]
},
{
 "cell_type": "code",
 "execution_count": 210,
 "id": "52634e53-de94-4cd6-af22-4f1292a132e9",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "{'orange'}"
    ]
   },
   "execution_count": 210,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "s1 & s2"
 ]
},
{
 "cell_type": "code",
 "execution_count": 1,
 "id": "4d40ef80-02e2-4ade-be84-a55bd40184c9",
 "metadata": {},
 "outputs": [],
 "source": [
  "# Difference : Return the elements that is present in first set and not in second set."
 ]
},
```

```
{
 "cell_type": "code",
 "execution_count": 5,
 "id": "e51ce77b-2648-42e6-9dac-8779f658ddbe",
 "metadata": {},
 "outputs": [],
 "source": [
  "s1 = {\"apple\",\"banana\",\"orange\",\"milk\",\"pen\"}"
 ]
},
{
 "cell_type": "code",
 "execution_count": 7,
 "id": "ccb21608-91b6-4fec-bc0d-2a3c73af3540",
 "metadata": {},
 "outputs": [],
 "source": [
  "s2 = {\"mobile\",\"bottle\",\"orange\",}"
 ]
},
{
 "cell_type": "code",
 "execution_count": 9,
 "id": "2d22c27e-ac15-439a-a8f5-9cc4b0b0fefb",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "{'apple', 'banana', 'milk', 'pen'}"
    ]
   },
   "execution_count": 9,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "s1 - s2"
 ]
},
{
 "cell_type": "code",
 "execution_count": 11,
 "id": "ba08fb17-d84a-48d1-923f-dbacef3954de",
 "metadata": {},
 "outputs": [],
 "source": [
  "# symmetric differences"
 ]
},
{
 "cell_type": "code",
 "execution_count": 13,
 "id": "350dab1b-ed55-442b-9feb-992abddc0aec",
 "metadata": {},
 "outputs": [],
 "source": [
```

```
 "s1 = {\"apple\",\"banana\",\"orange\",\"milk\",\"pen\"}"
]
},
{
"cell_type": "code",
"execution_count": 15,
"id": "2630b991-f9c4-46ca-bf61-c88fd796df7b",
"metadata": {},
"outputs": [],
"source": [
 "s2 = {\"mobile\",\"bottle\",\"orange\",}"
]
},
{
"cell_type": "code",
"execution_count": 17,
"id": "b2363623-4214-4b55-a5ce-4aabf60a6d41",
"metadata": {},
"outputs": [
 {
  "data": {
   "text/plain": [
    "{'apple', 'banana', 'bottle', 'milk', 'mobile', 'pen'}"
   ]
  },
  "execution_count": 17,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "s1 ^ s2"
]
},
{
"cell_type": "code",
"execution_count": 19,
"id": "90e3bbab-685a-4642-961d-b407cee48479",
"metadata": {},
"outputs": [],
"source": [
 "# Frozen sets : Immutable version of set, can not be added or removed any new element."
]
},
{
"cell_type": "code",
"execution_count": 21,
"id": "5721d61f-c9ea-4bd2-a611-b4c54bbd88cc",
"metadata": {},
"outputs": [],
"source": [
 "s = {1,2,3,4,5}"
]
},
{
"cell_type": "code",
"execution_count": 23,
"id": "811511bb-d53d-44c8-bbe6-54285571c2d1",
```

```
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "{1, 2, 3, 4, 5}"
    ]
   },
   "execution_count": 23,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "s"
 ]
},
{
 "cell_type": "code",
 "execution_count": 25,
 "id": "1cb1d9b4-9f35-46fd-be10-96345f5e28a5",
 "metadata": {},
 "outputs": [],
 "source": [
  "s.add (500)"
 ]
},
{
 "cell_type": "code",
 "execution_count": 27,
 "id": "a7df9ceb-b60b-401f-839a-25f43bf25f29",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "{1, 2, 3, 4, 5, 500}"
    ]
   },
   "execution_count": 27,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "s"
 ]
},
{
 "cell_type": "code",
 "execution_count": 29,
 "id": "c9472e50-0863-4cfa-986c-4e37f874ed2a",
 "metadata": {},
 "outputs": [],
 "source": [
  "s = frozenset ([1,2,3,4,5])"
 ]
},
```

```
 {
  "cell_type": "code",
  "execution_count": 31,
  "id": "1f9475aa-392e-4381-b2a8-b3218b99b166",
  "metadata": {},
  "outputs": [
   {
    "data": {
     "text/plain": [
      "frozenset({1, 2, 3, 4, 5})"
     ]
    },
    "execution_count": 31,
    "metadata": {},
    "output_type": "execute_result"
   }
  ],
  "source": [
   "s"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 33,
  "id": "1a711688-5101-4f9c-9ccb-b6821f6ce076",
  "metadata": {},
  "outputs": [
   {
    "ename": "AttributeError",
    "evalue": "'frozenset' object has no attribute 'add'",
    "output_type": "error",
    "traceback": [
     "\u001b[1;31m---------------------------------------------------------------------------\u001b[0m",
     "\u001b[1;31mAttributeError\u001b[0m                            Traceback (most recent call last)",
     "Cell \u001b[1;32mIn[33], line 1\u001b[0m\n\u001b[1;32m----> 1\u001b[0m
s\u001b[38;5;241m.\u001b[39madd (\u001b[38;5;241m500\u001b[39m)\n",
     "\u001b[1;31mAttributeError\u001b[0m: 'frozenset' object has no attribute 'add'"
    ]
   }
  ],
  "source": [
   "s.add (500)"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 35,
  "id": "0977d9c6-1c8d-40d3-8c8c-a0a057b05588",
  "metadata": {},
  "outputs": [
   {
    "ename": "AttributeError",
    "evalue": "'frozenset' object has no attribute 'pop'",
    "output_type": "error",
    "traceback": [
     "\u001b[1;31m---------------------------------------------------------------------------\u001b[0m",
     "\u001b[1;31mAttributeError\u001b[0m                            Traceback (most recent call last)",
```

```
      "Cell \u001b[1;32mIn[35], line 1\u001b[0m\n\u001b[1;32m----> 1\u001b[0m
s\u001b[38;5;241m.\u001b[39mpop()\n",
      "\u001b[1;31mAttributeError\u001b[0m: 'frozenset' object has no attribute 'pop'"
     ]
    }
   ],
   "source": [
    "s.pop()"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "e446bda5-c7b7-49ec-ba90-3037d66fa61f",
   "metadata": {},
   "source": [
    "#Question(6)_ Discuss the use cases of tuples and sets in Python programming."
   ]
  },
  {
   "cell_type": "markdown",
   "id": "33fb9755-0941-4834-97a3-26fdc8620c6e",
   "metadata": {},
   "source": [
    "##Answer_Tuples are immutable and sets is mutable."
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 42,
   "id": "46ab37fe-bce2-41a3-907f-68349e2df128",
   "metadata": {},
   "outputs": [],
   "source": [
    "#use case of Tuples"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 44,
   "id": "431d19de-8626-4e19-aeff-84a08e3c414d",
   "metadata": {},
   "outputs": [],
   "source": [
    "employ_name = (\"Aannu\",\"Bunny\",\"Sumit\",\"Sameer\",\"Sujata\")"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 46,
   "id": "9151e278-2d1c-45b0-a8e0-74903a704fb3",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "tuple"
      ]
     },
```

```
  "execution_count": 46,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "type(employ_name)"
]
},
{
"cell_type": "code",
"execution_count": 48,
"id": "63abe865-d3ef-4b2b-8ce2-3828748b2b03",
"metadata": {},
"outputs": [
 {
  "data": {
   "text/plain": [
    "'Aannu'"
   ]
  },
  "execution_count": 48,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "employ_name[ 0 ]"
]
},
{
"cell_type": "code",
"execution_count": 52,
"id": "3aa2f022-98cf-45df-9371-7f3ce4731904",
"metadata": {},
"outputs": [
 {
  "data": {
   "text/plain": [
    "'Bunny'"
   ]
  },
  "execution_count": 52,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "employ_name[ -4 ]"
]
},
{
"cell_type": "code",
"execution_count": 56,
"id": "36316d3e-8c50-401d-b9fb-b9af07506653",
"metadata": {},
"outputs": [
 {
```

```
    "ename": "TypeError",
    "evalue": "'tuple' object does not support item assignment",
    "output_type": "error",
    "traceback": [
     "\u001b[1;31m---------------------------------------------------------------------------\u001b[0m",
     "\u001b[1;31mTypeError\u001b[0m                                 Traceback (most recent call last)",
     "Cell \u001b[1;32mIn[56], line 1\u001b[0m\n\u001b[1;32m----> 1\u001b[0m employ_name[
\u001b[38;5;241m0\u001b[39m ] \u001b[38;5;241m=\u001b[39m
\u001b[38;5;124m\"\u001b[39m\u001b[38;5;124msunny\u001b[39m\u001b[38;5;124m\"\u001b[39m\n",
     "\u001b[1;31mTypeError\u001b[0m: 'tuple' object does not support item assignment"
    ]
   }
  ],
  "source": [
   "employ_name[ 0 ] = \"sunny\"                    # Tuples are immutable,dont modify the data."
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 58,
  "id": "b2dbd390-9b48-434b-8da3-aa1b89d51467",
  "metadata": {},
  "outputs": [],
  "source": [
   "# use case of sets"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 82,
  "id": "f5387842-aa86-47b1-8b17-ba8345216c0b",
  "metadata": {},
  "outputs": [],
  "source": [
   "list 1 = [1,2,5,\"apple\",\"orange\",\"banana\",\"brinjal\",\"apple\",\"grapes\"]"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 90,
  "id": "d10b5940-cc71-4b81-928d-73fa4aa75b07",
  "metadata": {},
  "outputs": [],
  "source": [
   "s = set(list1)"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 92,
  "id": "f7f3c3b9-a791-4ee3-96da-b56ca98835ce",
  "metadata": {},
  "outputs": [
   {
    "data": {
     "text/plain": [
      "{1, 2, 5, 'apple', 'banana', 'brinjal', 'grapes', 'orange'}"
     ]
```

```
    },
    "execution_count": 92,
    "metadata": {},
    "output_type": "execute_result"
   }
  ],
  "source": [
   "s"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 98,
  "id": "a4d334f4-04e8-4cda-a98f-6e4bbdb83d98",
  "metadata": {},
  "outputs": [
   {
    "data": {
     "text/plain": [
      "set"
     ]
    },
    "execution_count": 98,
    "metadata": {},
    "output_type": "execute_result"
   }
  ],
  "source": [
   "type (s)"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 102,
  "id": "4a9472eb-1a08-4d4b-9a6e-405197680618",
  "metadata": {},
  "outputs": [
   {
    "ename": "TypeError",
    "evalue": "'set' object is not subscriptable",
    "output_type": "error",
    "traceback": [
     "\u001b[1;31m---------------------------------------------------------------------------\u001b[0m",
     "\u001b[1;31mTypeError\u001b[0m                                 Traceback (most recent call last)",
     "Cell \u001b[1;32mIn[102], line 1\u001b[0m\n\u001b[1;32m----> 1\u001b[0m s[\u001b[38;5;241m0\u001b[39m]\n",
     "\u001b[1;31mTypeError\u001b[0m: 'set' object is not subscriptable"
    ]
   }
  ],
  "source": [
   "s[0]                          # set is unordered so indexing will not work"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 104,
  "id": "be95da58-f3be-4e10-8ecc-7ed43af85ecd",
```

```
 "metadata": {},
 "outputs": [],
 "source": [
  "s.add(100)"
 ]
},
{
 "cell_type": "code",
 "execution_count": 108,
 "id": "8cc05494-707b-4677-86c2-b3fd0dcd5cb7",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "{1, 100, 2, 5, 'apple', 'banana', 'brinjal', 'grapes', 'orange'}"
    ]
   },
   "execution_count": 108,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "s                                    # not compulsary that it will add to the last of set"
 ]
},
{
 "cell_type": "code",
 "execution_count": 110,
 "id": "3ee4b292-19b6-4ab3-9fbe-540cbc164394",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "1"
    ]
   },
   "execution_count": 110,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "s.pop()"
 ]
},
{
 "cell_type": "code",
 "execution_count": 112,
 "id": "b168cb03-3cc6-438e-b721-3eedc58fb3e0",
 "metadata": {},
 "outputs": [],
 "source": [
  "s.remove (5)"
 ]
},
```

```
{
 "cell_type": "code",
 "execution_count": 114,
 "id": "bc95c8c7-39eb-4077-a522-9cccd6f951d9",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "{100, 2, 'apple', 'banana', 'brinjal', 'grapes', 'orange'}"
    ]
   },
   "execution_count": 114,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "s"
 ]
},
{
 "cell_type": "code",
 "execution_count": 120,
 "id": "3aab1478-1b84-4e26-8a53-80cf1c66b15e",
 "metadata": {},
 "outputs": [],
 "source": [
  "s.update ([\"pooja\"])"
 ]
},
{
 "cell_type": "code",
 "execution_count": 122,
 "id": "58acd2d4-944e-45ac-9196-57185fb877fd",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "{100, 2, 'apple', 'banana', 'brinjal', 'grapes', 'orange', 'pooja'}"
    ]
   },
   "execution_count": 122,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "s"
 ]
},
{
 "cell_type": "code",
 "execution_count": 124,
 "id": "21f71d17-bb8d-45db-9b79-51b7dbd2d683",
 "metadata": {},
 "outputs": [],
```

```
 "source": [
  "s.clear()"
 ]
},
{
 "cell_type": "code",
 "execution_count": 126,
 "id": "2b97ba89-cf56-4c03-94ec-5623c5679310",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "set()"
    ]
   },
   "execution_count": 126,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "s"
 ]
},
{
 "cell_type": "code",
 "execution_count": 128,
 "id": "7ad6ea79-d02e-4501-96fb-f61b6cd44539",
 "metadata": {},
 "outputs": [],
 "source": [
  "del s"
 ]
},
{
 "cell_type": "code",
 "execution_count": 130,
 "id": "32e8742d-aed1-4ce8-ba7a-1afa81de7257",
 "metadata": {},
 "outputs": [
  {
   "ename": "NameError",
   "evalue": "name 's' is not defined",
   "output_type": "error",
   "traceback": [
    "\u001b[1;31m---------------------------------------------------------------------------\u001b[0m",
    "\u001b[1;31mNameError\u001b[0m                                 Traceback (most recent call last)",
    "Cell \u001b[1;32mIn[130], line 1\u001b[0m\n\u001b[1;32m----> 1\u001b[0m s\n",
    "\u001b[1;31mNameError\u001b[0m: name 's' is not defined"
   ]
  }
 ],
 "source": [
  "s"
 ]
},
{
```

```
  "cell_type": "code",
  "execution_count": null,
  "id": "b2491f3a-595f-4e51-b5c2-6f4d0ffff24b",
  "metadata": {},
  "outputs": [],
  "source": []
 },
 {
  "cell_type": "code",
  "execution_count": 132,
  "id": "adea8d62-6ab3-4f7b-b2a6-e4fcc438f1b6",
  "metadata": {},
  "outputs": [
   {
    "ename": "NameError",
    "evalue": "name 's' is not defined",
    "output_type": "error",
    "traceback": [
     "\u001b[1;31m---------------------------------------------------------------------------\u001b[0m",
     "\u001b[1;31mNameError\u001b[0m                                 Traceback (most recent call last)",
     "Cell \u001b[1;32mIn[132], line 1\u001b[0m\n\u001b[1;32m----> 1\u001b[0m 
s\u001b[38;5;241m.\u001b[39mremove(\u001b[38;5;241m100\u001b[39m)\n",
     "\u001b[1;31mNameError\u001b[0m: name 's' is not defined"
    ]
   }
  ],
  "source": [
   "s.remove(100)"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 134,
  "id": "c771151b-ae0a-4e74-a72d-c2e43867371c",
  "metadata": {},
  "outputs": [],
  "source": [
   "s = {1, 100, 2, 5, 'apple', 'banana', 'brinjal', 'grapes', 'orange'}"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 136,
  "id": "adb23067-5149-49d7-9486-3e6f415bf745",
  "metadata": {},
  "outputs": [],
  "source": [
   "s.remove (100)"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 138,
  "id": "c1bc843b-a6f6-4a15-bf6d-6e7efc5bb042",
  "metadata": {},
  "outputs": [
   {
    "data": {
```

```
    "text/plain": [
     "{1, 2, 5, 'apple', 'banana', 'brinjal', 'grapes', 'orange'}"
    ]
   },
   "execution_count": 138,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "s"
 ]
},
{
 "cell_type": "code",
 "execution_count": 146,
 "id": "9db0ebf5-2113-4866-943a-45d4f6432a54",
 "metadata": {},
 "outputs": [],
 "source": [
  "s.discard(0)"
 ]
},
{
 "cell_type": "code",
 "execution_count": 148,
 "id": "332759bb-2add-420d-a843-165c6f038b8d",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "{1, 2, 5, 'apple', 'banana', 'brinjal', 'grapes', 'orange'}"
    ]
   },
   "execution_count": 148,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "s"
 ]
},
{
 "cell_type": "code",
 "execution_count": 150,
 "id": "cfda8c02-73a9-43ac-865f-a0c74f73304a",
 "metadata": {},
 "outputs": [],
 "source": [
  "s.discard(1)"
 ]
},
{
 "cell_type": "code",
 "execution_count": 152,
 "id": "1bfca619-e43e-4ef3-9a43-919e01b0410d",
```

```
  "metadata": {},
  "outputs": [
   {
    "data": {
     "text/plain": [
      "{2, 5, 'apple', 'banana', 'brinjal', 'grapes', 'orange'}"
     ]
    },
    "execution_count": 152,
    "metadata": {},
    "output_type": "execute_result"
   }
  ],
  "source": [
   "s"
  ]
 },
 {
  "cell_type": "markdown",
  "id": "032688c8-0c8a-4b9a-b771-b60becf81f8f",
  "metadata": {},
  "source": [
   "#Question(7)_ Describe how to add, modify, and delete items in a dictionary with examples."
  ]
 },
 {
  "cell_type": "markdown",
  "id": "779d2815-8f1c-4dab-81ea-b9d38e690e7e",
  "metadata": {},
  "source": [
   "##Answer_'''Dictionary is a data structure that stores data as key value pair.\n",
   "          Unordered collections: Elements are not stored in a specific order.\n",
   "          Unique key-value pairs: Each key acts as a unique identifier for retrieving an associated value.\n",
   "          Flexible data: Keys and values can be of various data types (strings, numbers, lists, even other dictionaries).'''"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 159,
  "id": "200e127f-ff7a-4df0-af99-976a69192813",
  "metadata": {},
  "outputs": [],
  "source": [
   "d = {}"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 161,
  "id": "9f4c48a0-fd9a-4d62-9dc7-4e17e1e4f9cd",
  "metadata": {},
  "outputs": [
   {
    "data": {
     "text/plain": [
      "dict"
     ]
```

```
   },
   "execution_count": 161,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "type (d)"
 ]
},
{
 "cell_type": "code",
 "execution_count": 191,
 "id": "2c2d4f76-c214-46ef-bb72-aa92457ed246",
 "metadata": {},
 "outputs": [],
 "source": [
  "d = {\"name\" : \"Ansul\",\"email id\" : \"ansul123@gmail.com\",\"contact no.\" : 12345}"
 ]
},
{
 "cell_type": "code",
 "execution_count": 171,
 "id": "c84b003a-ddb8-4805-9c0b-8792cb8e038d",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "{'name': 'Ansul', 'email id': 'ansul123@gmail.com', 'contact no.': 12345}"
    ]
   },
   "execution_count": 171,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "d"
 ]
},
{
 "cell_type": "code",
 "execution_count": 193,
 "id": "e96a490c-cd76-4006-89b0-c3d2c18a340a",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "dict"
    ]
   },
   "execution_count": 193,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
```

```
   "source": [
    "type(d)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 205,
   "id": "9fe49ac7-4d16-46f8-ae20-4b1e61d13fd3",
   "metadata": {},
   "outputs": [],
   "source": [
    "d1 = {\"address\" : \"abcdefgh\"}"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 207,
   "id": "a3fc56b1-077c-4aea-9cef-2cf79af932cd",
   "metadata": {},
   "outputs": [],
   "source": [
    "d.update (d1)                    # Add: Use direct assignment (dictionary_name[key] = value) or .update()
method."
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 209,
   "id": "3297f068-bc8f-4216-956a-00527a45ecc4",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "{'name': 'Ansul',\n",
       " 'email id': 'ansul123@gmail.com',\n",
       " 'contact no.': 12345,\n",
       " 'address': 'abcdefgh'}"
      ]
     },
     "execution_count": 209,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "d"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 215,
   "id": "7aa25a13-be82-44fa-982d-1d2c1db6b7e0",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
```

```
    "'Ansul'"
   ]
  },
  "execution_count": 215,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "d [\"name\"]"
]
},
{
"cell_type": "code",
"execution_count": 217,
"id": "2aee21b4-b1a9-49b0-a321-1e6d089e409a",
"metadata": {},
"outputs": [],
"source": [
 "d [\"name\"] = \"suman\""
]
},
{
"cell_type": "code",
"execution_count": 219,
"id": "dbdf9c0a-de3c-4c30-a397-fd28ff06a5c9",
"metadata": {},
"outputs": [
 {
  "data": {
   "text/plain": [
    "{'name': 'suman',\n",
    " 'email id': 'ansul123@gmail.com',\n",
    " 'contact no.': 12345,\n",
    " 'address': 'abcdefgh'}"
   ]
  },
  "execution_count": 219,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "d"
]
},
{
"cell_type": "code",
"execution_count": null,
"id": "4aea84f3-8cd7-482f-a252-532004c32be2",
"metadata": {},
"outputs": [],
"source": [
 "Remove: Use del dictionary_name[key], .pop(key), or .popitem()."
]
},
{
"cell_type": "code",
```

```
 "execution_count": 221,
 "id": "ee6472f3-71a3-414c-993b-3632a6478da8",
 "metadata": {},
 "outputs": [],
 "source": [
  "del d [\"address\"]                                    "
 ]
},
{
 "cell_type": "code",
 "execution_count": 231,
 "id": "36924009-42a5-4ea7-a22f-8b05dfd2033e",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "{'name': 'suman', 'email id': 'ansul123@gmail.com', 'contact no.': 12345}"
    ]
   },
   "execution_count": 231,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "d"
 ]
},
{
 "cell_type": "code",
 "execution_count": 237,
 "id": "9a4540c2-4321-4cc8-a712-fbaf98b6c30b",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "'suman'"
    ]
   },
   "execution_count": 237,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "d.pop(\"name\")"
 ]
},
{
 "cell_type": "code",
 "execution_count": 239,
 "id": "bd9b05b2-679e-4c34-a364-73ba4c5c051f",
 "metadata": {},
 "outputs": [
  {
   "data": {
```

```
      "text/plain": [
       "{'email id': 'ansul123@gmail.com', 'contact no.': 12345}"
      ]
     },
     "execution_count": 239,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "d"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "b2230d41-ba8d-497a-a33a-1c7285ce13ec",
   "metadata": {},
   "source": [
    "#Question(8)_ Discuss the importance of dictionary keys being immutable and provide examples."
   ]
  },
  {
   "cell_type": "markdown",
   "id": "ccf2b677-2853-416c-83be-eb056e6190b5",
   "metadata": {},
   "source": [
    "##Answer_'''Python dictionaries allow us to associate a value to a unique key, and then to quickly access this value. \n",
    "          It's a good idea to use them whenever we want to find a certain Python object.\n",
    "          We can also use lists for this scope, but they are much slower than dictionaries.\n",
    "          The hash table implementation of dictionaries uses a hash value calculated from the key value to find the key. \n",
    "          If the key were a mutable object, its value could change, and thus its hash could also change.Thats why dictionary keys are unique and immutable.'''"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 5,
   "id": "9da4058a-d812-49a9-b289-f92a6df176f7",
   "metadata": {},
   "outputs": [],
   "source": [
    "d = {\"name\" : \"pooja\",\"email id\" : \"ansul123@gmail.com\",\"contact no.\" : 12345}"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 7,
   "id": "9054a750-1653-4bc6-8249-af1455c08936",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "{'name': 'pooja', 'email id': 'ansul123@gmail.com', 'contact no.': 12345}"
      ]
     },
```

```
  "execution_count": 7,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "d"
]
},
{
"cell_type": "code",
"execution_count": 9,
"id": "3cdefeb9-b263-4c8a-a944-21bff990fcae",
"metadata": {},
"outputs": [],
"source": [
 "d [\"name\"] = \"sameer\""
]
},
{
"cell_type": "code",
"execution_count": 11,
"id": "6ba4436e-57eb-452a-88aa-cb0aeef5a504",
"metadata": {},
"outputs": [
 {
  "data": {
   "text/plain": [
    "{'name': 'sameer', 'email id': 'ansul123@gmail.com', 'contact no.': 12345}"
   ]
  },
  "execution_count": 11,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "d"
]
},
{
"cell_type": "code",
"execution_count": 20,
"id": "2010fcab-87ed-4a3c-90a7-af712111cd14",
"metadata": {},
"outputs": [],
"source": [
 "d = {1 : \"abc\"}                    # integers as key"
]
},
{
"cell_type": "code",
"execution_count": 22,
"id": "2203850f-0b39-470a-828b-49f90c4abc68",
"metadata": {},
"outputs": [
 {
  "data": {
```

```
    "text/plain": [
     "{1: 'abc'}"
    ]
   },
   "execution_count": 22,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "d"
 ]
},
{
 "cell_type": "code",
 "execution_count": 24,
 "id": "736c5af3-a900-44d7-a012-f5bad9a7b4b4",
 "metadata": {},
 "outputs": [],
 "source": [
  "d = {1.5 : \"abc\"}                  # float as key"
 ]
},
{
 "cell_type": "code",
 "execution_count": 26,
 "id": "ac40aecf-e009-4f2f-8df0-d3e003dc316c",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "{1.5: 'abc'}"
    ]
   },
   "execution_count": 26,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "d"
 ]
},
{
 "cell_type": "code",
 "execution_count": 28,
 "id": "20d59f92-d90d-4a01-865c-f4932e3a8697",
 "metadata": {},
 "outputs": [],
 "source": [
  "d = {\"True\" : \"abc\"}               # boolean value also can be used as a key"
 ]
},
{
 "cell_type": "code",
 "execution_count": 30,
 "id": "b51ffa33-6273-4eea-bc16-f8bd9c390b64",
```

```
  "metadata": {},
  "outputs": [
   {
    "data": {
     "text/plain": [
      "{'True': 'abc'}"
     ]
    },
    "execution_count": 30,
    "metadata": {},
    "output_type": "execute_result"
   }
  ],
  "source": [
   "d"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 32,
  "id": "f64385f9-f07a-4488-85b6-fe764cc0369f",
  "metadata": {},
  "outputs": [
   {
    "ename": "TypeError",
    "evalue": "unhashable type: 'list'",
    "output_type": "error",
    "traceback": [
     "\u001b[1;31m---------------------------------------------------------------------------\u001b[0m",
     "\u001b[1;31mTypeError\u001b[0m                                 Traceback (most recent call last)",
     "Cell \u001b[1;32mIn[32], line 1\u001b[0m\n\u001b[1;32m----> 1\u001b[0m d
\u001b[38;5;241m=\u001b[39m
{[\u001b[38;5;241m1\u001b[39m,\u001b[38;5;241m2\u001b[39m,\u001b[38;5;241m3\u001b[39m] :
\u001b[38;5;124m\"\u001b[39m\u001b[38;5;124mabc\u001b[39m\u001b[38;5;124m\"\u001b[39m}\n",
     "\u001b[1;31mTypeError\u001b[0m: unhashable type: 'list'"
    ]
   }
  ],
  "source": [
   "d = {[1,2,3] : \"abc\"}                # List can not be used as a key"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 34,
  "id": "361887a2-8b70-4354-9106-7cce24fbd9a0",
  "metadata": {},
  "outputs": [],
  "source": [
   "d = {(1,2,3) : \"abc\"}                 # only string and numbers can be used as a key of the dictionary."
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 36,
  "id": "131b03dc-24a9-4086-9367-01607ff32da1",
  "metadata": {},
  "outputs": [
```

```
  {
   "data": {
    "text/plain": [
     "{(1, 2, 3): 'abc'}"
    ]
   },
   "execution_count": 36,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "d"
 ]
},
{
 "cell_type": "code",
 "execution_count": 38,
 "id": "515ba302-4f8c-42ae-b514-df0af27911e4",
 "metadata": {},
 "outputs": [],
 "source": [
  "d = {\"name\" : \"pooja\",\"email id\" : \"ansul123@gmail.com\",\"contact no.\" : 12345}"
 ]
},
{
 "cell_type": "code",
 "execution_count": 44,
 "id": "a762a0be-cec8-4aa4-a433-9a7a2d13281e",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "'pooja'"
    ]
   },
   "execution_count": 44,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "d [\"name\"]"
 ]
},
{
 "cell_type": "code",
 "execution_count": 46,
 "id": "01efc814-2136-4ff9-9eaf-dacd73afb190",
 "metadata": {},
 "outputs": [
  {
   "ename": "KeyError",
   "evalue": "'pooja'",
   "output_type": "error",
   "traceback": [
    "\u001b[1;31m---------------------------------------------------------------------------\u001b[0m",
```

```
      "\u001b[1;31mKeyError\u001b[0m                            Traceback (most recent call last)",
      "Cell \u001b[1;32mIn[46], line 1\u001b[0m\n\u001b[1;32m----> 1\u001b[0m d
[\u001b[38;5;124m\"\u001b[39m\u001b[38;5;124mpooja\u001b[39m\u001b[38;5;124m\"\u001b[39m]\n",
      "\u001b[1;31mKeyError\u001b[0m: 'pooja'"
     ]
    }
   ],
   "source": [
    "d [\"pooja\"]                    # Only can access value using key,vice-versa is not possible.\n",
    "                               # keys are immutable."
   ]
  }
 ],
 "metadata": {
  "kernelspec": {
   "display_name": "Python 3 (ipykernel)",
   "language": "python",
   "name": "python3"
  },
  "language_info": {
   "codemirror_mode": {
    "name": "ipython",
    "version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.12.4"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 5
}
```