RAMAIAH UNIVERSITY OF APPLIED SCIENCES

ASSIGNMENT

Course Code	CSC201A
Course Name	Discrete Mathematics -1
Programme	B. Tech
Department	Computer Science & Engineering
Faculty	Faculty of Engineering Technology

Name of the Student	SUBHENDU MAJI
Reg. No	18ETCS002121
Semester/Year	3RD / 2019
Course Leader/s	Ms Sahana P. Shankar

Declaration Sheet			
Student Name	SUBHENDU MAJI		
Reg. No	18ETCS002121		
Programme	B. Tech	Semester/Year	3rd / 2019
Course Code	CSC201A		
Course Title	Discrete Mathematics -1		
Course Date		To	
Course Leader	Ms Sahana P. Shankar		

Declaration

The assignment submitted herewith is a result of my own investigations and that I have conformed to the guidelines against plagiarism as laid out in the Student Handbook. All sections of the text and results, which have been obtained from other sources, are fully referenced. I understand that cheating and plagiarism constitute a breach of University regulations and will be dealt with accordingly.

Signature of the Student		Date	
Submission date stamp (by Examination & Assessment Section)			

Signature of the Course Leader and date	Signature of the Reviewer and date

Contents

Engineering and Technology			
Ramaiah University of Applied Sciences			
Department	Computer Science and Engineering	Programme	B. Tech.
Semester/Batch	3^{rd}/2019		
Course Code	CSC201A	Course Title	Discrete Mathematics-1
Course Leader(s)	Ms Sahana P. Shankar and Ms.Supriya		

Questions	Marking Scheme		Marks		
			Max Marks	First Examiner Marks	Moderator
1	1.1	Development of a Python program with comments to create sets and perform the specified operations	5		
	1.2	Illustration using Venn diagrams	2		
		Question 1 Max Marks	**7**		
2	2.1	Justification of the statement with appropriate reasoning	2		
	2.2	Solution to the example problem	1		
		Question 2 Max Marks	**3**		
		Total Assignment Marks	**10**		

Course Marks Tabulation				
Question	**First Examiner**	**Remarks**	**Moderator**	**Remarks**
1				
2				
Marks (Max 10)				

Signature of First Examiner **Signature of Moderator**

Question No. 1

Solution to Question No. 1:

1.1 Development of a Python program with comments to create sets and perform the specified operations

inputing values of universal set, set A and set B

In [1]:
```
U=set(list(input("Enter Universal set values: ").split(' ')))
A=set(list(input("Enter set A values: ").split(' ')))
B=set(list(input("Enter set B values:").split(' ')))
```

```
Enter Universal set values: 1 2 3 4 5 6 a b c d e
Enter set A values: 1 3 4 6 a c e
Enter set B values:2 3 5 6 b c d
```

In [2]:
```
print("A = ",A)
print("B = ",B)
print("U = ",U)
```

```
A =  {'a', '1', '3', '4', '6', 'c', 'e'}
B =  {'d', '5', '3', '2', 'b', '6', 'c'}
U =  {'a', '1', 'd', '5', '3', '2', 'b', '4', '6', 'c', 'e'}
```

sets operations using inbuilt commands

In [3]:
```
# union
print("Union set :", A | B)

# intersection
print("Intersection set :", A & B)

# difference
print("Difference set:", A - B)

# compliment
print("Compliment set :", U - A)
```

```
Union set : {'a', '1', 'd', '5', '3', '2', 'b', '4', '6', 'c', 'e'}
Intersection set : {'6', '3', 'c'}
Difference set: {'a', '1', '4', 'e'}
Compliment set : {'2', 'd', 'b', '5'}
```

functions for doing set operations from scratch (without using inbuilt commands)

In [4]:
```
def union(A,B): #function for doing union of sets A and B
    s=[]
    for x in A:
        s.append(x)
    for x in B:
        s.append(x)
    return set(s)
```

In [5]:
```
def intersection(A,B): #function for intersection of sets A and B
    s=[]
    for x in A:
        if x in B:
            s.append(x)
    return set(s)
```

In [6]:
```
def difference(A,B): #function for doing difference of sets a and b
    s=[]
    for x in A:
        if x not in B:
            s.append(x)
    return set(s)
```

In [7]:
```
def compliment(U,A): #function for doing compliment of set a
    return difference(U,A)
```

calling function for performing set operations

In [11]:
```
# union
print("Union set :", union(A,B))

# intersection
print("Intersection set :", intersection(A,B))

# difference
print("Difference set:", difference(A,B))

# compliment
print("Compliment set :", compliment(U,A))
```

```
Union set : {'a', '1', 'd', '5', '3', '2', 'b', '4', '6', 'c', 'e'}
Intersection set : {'6', '3', 'c'}
Difference set: {'a', '1', '4', 'e'}
Compliment set : {'2', 'd', 'b', '5'}
```

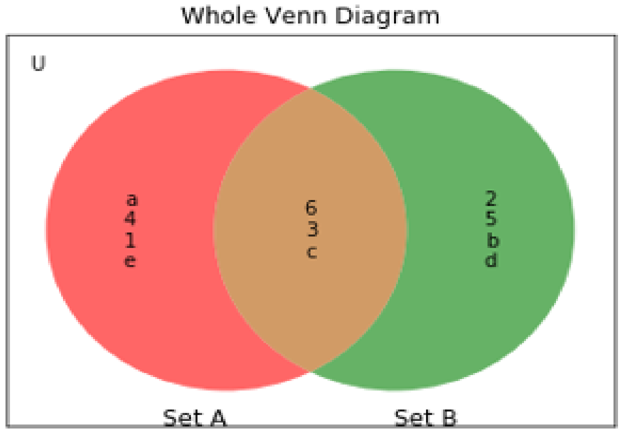
1.2 Illustration using Venn diagrams

importing matplotlib for venn diagram

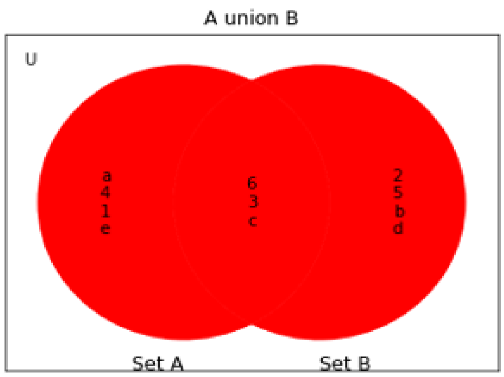
In [9]:
```
import matplotlib_venn as vplt
from matplotlib import pyplot as plt
```

In [10]:
```
plt.title('Whole Venn Diagram')
v = vplt.venn2([A,B], ('Set A', 'Set B'),alpha=0.6)

v.get_label_by_id('10').set_text('\n'.join(A-B))
v.get_label_by_id('11').set_text('\n'.join(A&B))
v.get_label_by_id('01').set_text('\n'.join((B-A)))
plt.gca().set_facecolor('white')
plt.annotate('U',xy=(-0.7,0.45))
plt.gca().set_axis_on()
```

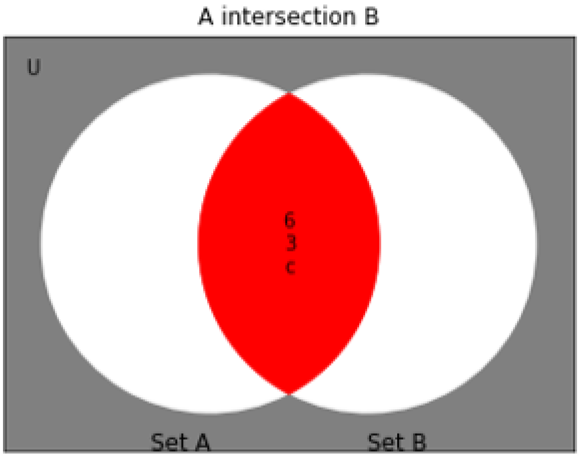


In [11]:
```
plt.title('A union B')
v = vplt.venn2([A,B], ('Set A', 'Set B'),set_colors=('r','r'),alpha=1)
v.get_patch_by_id('11').set_color('r')
v.get_label_by_id('10').set_text('\n'.join(A-B))
v.get_label_by_id('11').set_text('\n'.join(A&B))
v.get_label_by_id('01').set_text('\n'.join(B-A))
plt.gca().set_facecolor('w')
plt.annotate('U',xy=(-0.7,0.45))
plt.gca().set_axis_on()
```



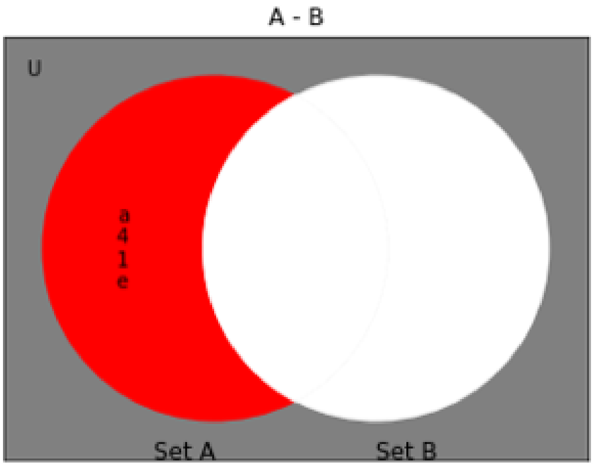
In [12]:

```
plt.title('A intersection B')
v = vplt.venn2([A,B], ('Set A', 'Set B'),set_colors=('w','w'),alpha=1)
v.get_patch_by_id('11').set_color('red')
v.get_label_by_id('10').set_text('')
v.get_label_by_id('11').set_text('\n'.join(A&B))
v.get_label_by_id('01').set_text('')
plt.gca().set_facecolor('grey')
plt.annotate('U',xy=(-0.7,0.45))
plt.gca().set_axis_on()
```



In [13]:

```
plt.title('A - B')
v = vplt.venn2([A,B], ('Set A', 'Set B'),set_colors=('r','w'),alpha=1)
v.get_patch_by_id('11').set_color('w')
v.get_label_by_id('10').set_text('\n'.join(A-B))
v.get_label_by_id('11').set_text('')
v.get_label_by_id('01').set_text('')
plt.gca().set_facecolor('grey')
plt.annotate('U',xy=(-0.7,0.45))
plt.gca().set_axis_on()
```



Question No. 2

Solution to Question No. 2:

2.1 Justification of the statement with appropriate reasoning

"Mathematical Induction can be applied to prove that the cardinality of a power set is 2^n, where n is the cardinality of the set"

Yes, the statement is True.
To Prove: For any finite set S, if $|S| = n$, then S has $2n$ subsets.
Proof:
Let P(n) be the predicate "A set with cardinality n has 2n subsets".
Basis step:
$P(0)$ is true, because the set with cardinality 0 (the empty set) has 1 subset (itself) and $2^0 = 1$.
Inductive step: To Prove $\boldsymbol{P(k)} \rightarrow \boldsymbol{P(k+1)}$

That is, prove that if a set with k elements has 2k subsets, then a set with $k+1$ elements has $2k+1$ subsets.
Proof
Assume that for an arbitrary k, any set with cardinality k has $2k$ subsets.
Let T be a set such that $|T| = k+1$.
Enumerate the elements of T: $T = \{ e_1, e_2, e_3 \dots \dots e_k, e_{k+1}\}$
Let $S = \{ e_1, e_2, e_3 \dots \dots e_k\}$
Then $|S| = k$, so S has 2^k subsets, according to the inductive hypothesis.

Note that $T = S \cup \{e_{k+1}\}$, so every subset of S is also a subset of T.

Any subset of T either contains the element e_{k+1}, or it doesn't contain e_{k+1}.
If a subset of T doesn't contain e_{k+1}, then it is also a subset of S, and there are 2^k of those subsets.

On the other hand, if a subset of T does contain the element e_{k+1}, then that subset is formed by including e_{k+1} in one of the 2^k subsets of S, so T has 2^k subsets containing e_{k+1}.

We have shown that T has 2^k subsets containing e_{k+1}, and another 2^k subsets not containing e_{k+1}, so the total number of subsets of T is $\mathbf{2^k + 2^k = (2)2^k = 2^{k+1}}$**.**

2.2 Solution to the example problem

Example.
Let S be a set,
$\boldsymbol{S} = \{\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}\}$
Cardinality of S is given by $|S|$, hence **|S|=4**

Subsets of S:

$$Subsets\ of\ cardinality\ 0 = \emptyset$$
$$Subsets\ of\ cardinality\ 1 = \{1\}, \{2\}, \{3\}, \{4\}$$
$$Subsets\ of\ cardinality\ 2 = \{1,2\}, \{2,3\}, \{3,4\}, \{2,4\}, \{3,1\}, \{4,1\}$$
$$Subsets\ of\ cardinality\ 3 = \{1,2,3\}, \{1,2,4\}, \{1,3,4\}, \{2,3,4\}$$
$$Subsets\ of\ cardinality\ 4 = \{1,2,3,4\}$$

Hence the , $power\ set\ of\ X,\ P(X) =$
$\{ \emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1,2\}, \{2,3\}, \{3,4\}, \{2,4\}, \{3,1\}, \{4,1\}, \{1,2,3\}, \{1,2,4\}, \{1,3,4\}, \{2,3,4\}, \{1,2,3,4\} \}$
$Cardinality\ of\ |P(X)| = 16$
Therefore, the power set will be the set of all subsets
i.e. its cardinality is equal to total number of subsets, $16 = \mathbf{2^4}$.

Bibliography

- https://www.math.fsu.edu/~wooland/mad2104/proof/pf20.pdf
- https://stackoverflow.com/questions/54603761/is-it-possible-to-display-the-venn-diagram-within-a-universal-set

  For Python Source Code:
- https://github.com/subhendu17620/RUAS/blob/master/sem%2003/DM-1/DM-assignment01.ipynb

  https://github.com/subhendu17620