**CODE :**

#include<iostream>

using namespace std;

struct SLLNode\* createSLL(int cnt, struct SLLNode \*head);

void displaySLL(struct SLLNode \*head);

void A\_U\_B();

void A\_int\_B();

void A\_Min\_B();

void B\_Min\_A();

void U\_Min\_A\_U\_B();

struct SLLNode

{

char data;

struct SLLNode \*next;}

\*headU, \*headA, \*headB;

int main()

{

int i,no,m,v,b;

cout<<"\n\nHow many Sets to be stored: ";

cin>>no;

headU = headA = headB = NULL;

cout<<"Enter the number of students: ";

cin>>m;

cout<<"Enter the number of students who like Vanilla Ice-cream: ";

cin>>v;

cout<<"Enter the number of students who like Butterscotch Ice-cream: ";

cin>>b;

for(i=1; i<=no; i++)

{

if(i == 1)

{

cout<<"\n\nEnter "<<m<< " Students of SE Comp : ";

headU = createSLL(m, headU);

cout<<"\n";

displaySLL(headU);

}

if(i == 2)

{

cout<<"\n\nEnter "<<v<<" Students who like Vanilla Icecream: ";

headA = createSLL(v, headA);

cout<<"\n";

displaySLL(headA);

}

if(i == 3)

{

cout<<"\n\nEnter "<<b<<" Students who like Butterscotch Icecream: ";

headB = createSLL(b, headB);

cout<<"\n";

displaySLL(headB);

}

}

cout<<"\n\n Input Sets:------------------------";

cout<<"\n\n\t Set 'U': ";

displaySLL(headU);

cout<<"\n\n\t Set 'A': ";

displaySLL(headA);

cout<<"\n\n\t Set 'B': ";

displaySLL(headB);

cout<<"\n\n Output Sets:------------------------";

A\_U\_B();

A\_int\_B();

A\_Min\_B();

B\_Min\_A();

U\_Min\_A\_U\_B();

cout<<"\n\n";

return 0;

}

//.........................................................Function to create Linked List as Sets.

struct SLLNode\* createSLL(int cnt, struct SLLNode \*head)

{

int i;

struct SLLNode \*p, \*newNode;

for(i=0; i<cnt; i++)

{

newNode = new(struct SLLNode); // 1. DMA

cout<<"\n\t Enter Student Initial: "; // 2. Data & Address Assignment

cin>>newNode->data;

newNode->next = NULL;

if(head == NULL) // 3. Add node in the list

{

head = newNode;

p = head;

}

else

{

p->next = newNode;

p = p->next;

}

}

return head;

}

//...............................................Function to display Linked Lists as Sets.

void displaySLL(struct SLLNode \*head)

{

struct SLLNode \*p;

p = head;

while(p != NULL)

{

cout<<" "<<p->data;

p = p->next;

}

}

//................................................Function for Set A U B .

void A\_U\_B()

{

int i,j;

char a[10];

struct SLLNode \*p, \*q;

i = 0; //Index of Resultant Array

p = headA; //pointer to Set 'A'

q = headB; //pointer to Set 'B'

while(p != NULL && q != NULL)

{

if(p->data == q->data)

{

a[i] = p->data;

i++;

p = p->next;

q = q->next;

}

else

{

a[i] = p->data;

i++;

p = p->next;

}

}

if(p == NULL) //Set 'A' copied completely

{

while(q != NULL) //Copy remaining elements of Set 'B'

{

a[i] = q->data;

i++;

q = q->next;

}

}

if(q == NULL) //Set 'B' copied completely

{

while(p != NULL) //Copy remaining elements of Set 'A'

{

a[i] = p->data;

i++;

p = p->next;

}

}

cout<<"\n\n\t Set A U B: ";

for(j=0; j < i; j++)

cout<<" "<<a[j];

}

//................................................Function for Set A . B .

void A\_int\_B()

{

int i,j;

char a[10];

struct SLLNode \*p, \*q;

i = 0; //Index of Resultant Array

p = headA; //pointer to Set 'A'

while(p != NULL)

{

q = headB; //pointer to Set 'B'

while(q != NULL)

{

if(p->data == q->data)

{

a[i] = p->data;

i++;

}

q = q->next;

}

p = p->next;

}

cout<<"\n\n\t Set A . B: ";

for(j=0; j < i; j++)

cout<<" "<<a[j];

}

//................................................Function for Set A - B .

void A\_Min\_B()

{

int i,j,flag;

char a[10];

struct SLLNode \*p, \*q;

i = 0; //Index of Resultant Array

p = headA; //pointer to Set 'A'

while(p != NULL)

{

flag = 0;

q = headB; //pointer to Set 'B'

while(q != NULL)

{

if(p->data == q->data)

{

flag = 1;

}

q = q->next;

}

if(flag == 0)

{

a[i] = p->data;

i++;

}

p = p->next;

}

cout<<"\n\n\t Set A - B: ";

for(j=0; j < i; j++)

cout<<" "<<a[j];

}

//................................................Function for Set B - A.

void B\_Min\_A()

{

int i,j,flag;

char a[10];

struct SLLNode \*p, \*q;

i = 0; //Index of Resultant Array

q = headB; //pointer to Set 'B'

while(q != NULL)

{

flag = 0;

p = headA; //pointer to Set 'A'

while(p != NULL)

{

if(q->data == p->data)

{

flag = 1;

}

p = p->next;

}

if(flag == 0)

{

a[i] = q->data;

i++;

}

q = q->next;

}

cout<<"\n\n\t Set B - A: ";

for(j=0; j < i; j++)

cout<<" "<<a[j];

}

//.........................................Function for Set U - (A U B)

void U\_Min\_A\_U\_B()

{

int i,j,flag;

char a[10];

struct SLLNode \*p, \*q, \*r;

i = 0; //Index of Resultant Array

p = headU; //pointer to Set 'U'

while(p != NULL)

{

flag = 0;

q = headA; //pointer to Set 'A'

r = headB; //pointer to Set 'B'

while(q != NULL)

{

if(p->data == q->data)

{

flag = 1;

}

q = q->next;

}

while(r != NULL)

{

if(p->data == r->data)

{

flag = 1;

}

r = r->next;

}

if(flag == 0)

{

a[i] = p->data;

i++;

}

p = p->next;

}

cout<<"\n\n\t Set U - (A U B): ";

for(j=0; j < i; j++)

cout<<" "<<a[j];

}

**OUTPUT :**

How many Sets to be stored: 3

Enter the number of students: 7

Enter the number of students who like Vanilla Ice-cream: 4

Enter the number of students who like Butterscotch Ice-cream: 3

Enter 7 Students of SE Comp :

Enter Student Initial: a

Enter Student Initial: b

Enter Student Initial: c

Enter Student Initial: d

Enter Student Initial: e

Enter Student Initial: f

Enter Student Initial: g

a b c d e f g

Enter 4 Students who like Vanilla Icecream:

Enter Student Initial: a

Enter Student Initial: b

Enter Student Initial: c

Enter Student Initial: d

a b c d

Enter 3 Students who like Butterscotch Icecream:

Enter Student Initial: b

Enter Student Initial: c

Enter Student Initial: e

b c e

Input Sets:------------------------

Set 'U': a b c d e f g

Set 'A': a b c d

Set 'B': b c e

Output Sets:------------------------

Set A U B: a b c d e

Set A . B: b c

Set A - B: a d

Set B - A: e

Set U - (A U B): f g