**ROLL NO. 50**

**ASSIGNMENT NO. 3 : CRYPT-ARITHMETIC**

#include <stdlib.h>

#include <stdio.h>

#include <conio.h>

#include <string.h>

struct Node

{

char symbol;

int value;

struct Node \*next;

};

typedef struct Node node;

char str1[10],str2[10],str3[10];

node \*head=NULL;

int count=0;

int inList(node \*head,char a)

{

node \*p;

p = head;

while(p!=NULL)

{

if(p->symbol == a)

return 1;

p = p->next;

}

return 0;

}

node\* add\_to\_list(node\* head,char a)

{

node \*temp = (node\*)malloc(sizeof(node));

temp->symbol = a;

temp->value = 0;

temp->next = NULL;

if(head == NULL)

head = temp;

else

{

node \*p = head;

while(p->next!=NULL)

p = p->next;

p->next = temp;

}

return head;

}

int findValue(char a)

{

node \*p=head;

while(p!=NULL)

{

if(p->symbol == a)

return p->value;

p = p->next;

}

}

void checkSum()

{

int val1=0;

int val2=0;

int val3=0;

int m=1;

int i;

node \*p;

for(i=strlen(str1); i>=0; i--)

{

val1 = val1+m\*findValue(str1[i]);

m=m\*10;

}

//printf("\nValue1 = %d\n",val1);

m=1;

for(i=strlen(str2); i>=0; i--)

{

val2 = val2+m\*findValue(str2[i]);

m=m\*10;

}

//printf("\nValue2 = %d\n",val2);

m=1;

for(i=strlen(str3); i>=0; i--)

{

val3 = val3+m\*findValue(str3[i]);

m=m\*10;

}

//printf("\nValue3 = %d\n",val3);

if( (val1+val2) == val3 )

{

printf("\nSolution %d\n",count+1);

p = head;

while(p!=NULL)

{

printf("%c = %d\n",p->symbol,p->value);

p=p->next;

}

count++;

}

}

void findValues(node \*p,int assigned[])

{

int i;

if(p==NULL)

{

checkSum();

return;

}

for(i=0; i<10; i++)

{

if(assigned[i] == 0)

{

p->value = i;

assigned[i] = 1;

findValues(p->next,assigned);

assigned[i] = 0;

}

}

return;

}

node\* createList()

{

int i;

int length = strlen(str1);

//printf("I am in create List");

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

{

if( !inList(head,str1[i]) )

head = add\_to\_list(head,str1[i]);

}

length = strlen(str2);

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

{

if( !inList(head,str2[i]) )

head = add\_to\_list(head,str2[i]);

}

length = strlen(str3);

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

{

if( !inList(head,str3[i]) )

head = add\_to\_list(head,str3[i]);

}

return head;

}

int main()

{

node \*p;

int assigned[10]={0};

printf("Enter string 1 = ");

gets(str1);

printf("Enter string 2 = ");

gets(str2);

printf("Enter string 3 = ");

gets(str3);

head = createList();

printf("\nCurrent list = ");

p = head;

while(p!=NULL)

{

printf("%c->",p->symbol);

p = p->next;

}

//printf("\nhead->symbol = %c",head->symbol);

findValues(head,assigned);

//checkSum();

printf("\nTotal = %d\n",count);

}

**OUTPUT :**



