Q1:

```
#include <stdio.h>
#include <stdlib.h>
#include<string.h>
struct node
{
   int info;
   struct node *link;
};
struct node* insert_end(struct node *start,int info);
struct node* insert_beginning(struct node *start,int info);
void reverse(char a[],int i,int size)
{
   char temp;
   temp=a[i];
   a[i]=a[size-i];
   a[size-i]=temp;
   if(i==size/2){
      return;
   }
   reverse(a,i+1,size);
}
void tower(int n, char source, char dest, char aux)
{
   if (n == 1)
   {
      printf("\nMove disc 1 from %c to %c", source, dest);
      return;
   }
   tower(n - 1, source, aux, dest);
   printf("\nMove disk %d from %c to %c", n, source, dest);
   tower(n - 1, aux, dest, source);
}
int base(int n, int b1, int b2)
{
   if (n == 0)
      return 0;
   else
   {
      return (n % b2 + b1 * base(n / b2, b1, b2));
   }
}
int gcd(int a,int b)
{
   if(b>a)
   return gcd(b,a);
   if(b==0)
   return a;
   else
   return gcd(b,a%b);
}
struct node* createlist(struct node *start,int size)
{
   int info;
   printf("enter the number :");
   scanf("%d",&info);
   start=insert_beginning(start,info);
```

```
   for(int i=2;i<=size;i++)
   {
   printf("enter the number :");
   scanf("%d",&info);
   start=insert_end(start,info);
   }
   return start;

}
struct node* insert_beginning(struct node *start,int info)
{
   struct node *temp;
   temp=(struct node*)malloc(sizeof(struct node));
   temp->info=info;
   temp->link=start;
   start=temp;
   return start;
}
struct node* insert_end(struct node *start,int info)
{
   struct node *temp,*p;
   p=(struct node *) malloc(sizeof(struct node));
   temp=start;
   while(temp->link!=NULL)
      temp=temp->link;
   temp->link=p;
   p->info=info;
   p->link=NULL;
   return start;
}
void display(struct node *start)
{
   struct node *temp;
   temp=start;
   while(temp!=NULL)
   {
      printf("%d ",temp->info);
   }
}
void search(struct node *start,int key)
{
   if(start->info==key){
      printf("\nKey is found");
   }
   if(start->link==NULL)
   {
      return;
   }
   search(start->link,key);
}
int main()
{
   struct node *start;
   int n, b1, b2, choice, m,num1,num2,size,key,k;
   char a[40];
   while (1)
   {
      printf("\n1.Base Cnversion");
```

```
        printf("\n2.Tower Of Hanoi");
        printf("\n3.Greatest Common Divisor");
        printf("\n4.Reverse a string");
        printf("\n5.Search an element in a Linkedlist");
        printf("\n6.Exit");
        scanf("%d", &choice);
        switch (choice)
        {
        case 1:
            printf("\nEnter the number to be converted : ");
            scanf("%d", &n);
            printf("\nEnter the base of the number : ");
            scanf("%d", &b1);
            printf("\nEnter the base into to which the number is to be converted : ");
            scanf("%d", &b2);
            printf("\nResult : %d\n", base(n, b1, b2));
            break;
        case 2:
            printf("\nEnter the number of disks : ");
            scanf("%d", &m);
            printf("\nThe sequence of moves involved in the tower of Hanoi are : ");
            tower(m, 'A', 'C', 'B');
            break;
        case 3:
        printf("\nEnter the number 1 : ");
        scanf("%d",&num1);
        printf("\nEnter the number 2 : ");
        scanf("%d",&num2);
        printf("\nThe greatest commom divisor is : %d",gcd(num1,num2));
        break;
        case 4:
        printf("\nEnter a string : ");
        scanf("%s",a);
        size=strlen(a);
        reverse(a,0,size-1);
        printf("\nThe reversed string is %s",a);
        break;
        case 5:
        printf("\nEnter the number of nodes : ");
        scanf("%d",&k);
        start=createlist(start,k);
        printf("\nEnter the key to be search : ");
        scanf("%d",&key);
        search(start,key);
        break;
        case 6:
            exit(1);
        }
    }
    return 0;
}
```

OUTPUT:

1.Base Cnversion

2.Tower Of Hanoi

3.Greatest Common Divisor

4.Reverse a string

5.Search an element in a Linkedlist

6.Exit5

Enter the number of nodes : 3

enter the number :1

enter the number :2

enter the number :33

Enter the key to be search : 2

Key is found

1.Base Cnversion

2.Tower Of Hanoi

3.Greatest Common Divisor

4.Reverse a string

5.Search an element in a Linkedlist

6.Exit1

Enter the number to be converted : 10

Enter the base of the number : 10

Enter the base into to which the number is to be converted : 2

Result : 1010

1.Base Cnversion

2.Tower Of Hanoi

3.Greatest Common Divisor

4.Reverse a string

5.Search an element in a Linkedlist

6.Exit3

Enter the number 1 : 4

Enter the number 2 : 8

The greatest commom divisor is : 4

1.Base Cnversion

2.Tower Of Hanoi

3.Greatest Common Divisor

4.Reverse a string

5.Search an element in a Linkedlist

6.Exit2

Enter the number of disks : 3

The sequence of moves involved in the tower of Hanoi are :

Move disc 1 from A to C

Move disk 2 from A to B

Move disc 1 from C to B

Move disk 3 from A to C

Move disc 1 from B to A

Move disk 2 from B to C

Move disc 1 from A to C

1.Base Cnversion

2.Tower Of Hanoi

3.Greatest Common Divisor

4.Reverse a string

5.Search an element in a Linkedlist

6.Exit4

Enter a string : SIDDESH

The reversed string is HSEDDIS

1.Base Cnversion

2.Tower Of Hanoi

3.Greatest Common Divisor

4.Reverse a string

5.Search an element in a Linkedlist

6.Exit