**CODE:-**

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

#include <string.h>

int base\_dec(int, int, int \*, int \*);

void base\_rec(int, int);

void base\_conv(int num, int base\_n, int base\_c)

{

int count = 0, sum = 0;

if (base\_n != 10)

num = base\_dec(num, base\_n, &sum, &count);

base\_rec(num, base\_c);

}

int base\_dec(int num, int base\_n, int \*sum, int \*count)

{

if (num == 0)

return \*sum;

else

{

\*sum += ((num % 10) \* pow(base\_n, (\*count)++));

base\_dec(num / 10, base\_n, sum, count);

}

}

void base\_rec(int num, int base\_c)

{

if (num == 0)

return;

base\_rec(num / base\_c, base\_c);

int rem;

rem = num % base\_c;

if (rem > 9)

printf("%c", 'A' + rem - 9);

else

printf("%d", rem);

}

void towerh(int num, char from, char to, char temp)

{

if (num == 1)

{

printf("Move disk 1 from rod %c to rod %c.\n", from, to);

return;

}

towerh(num - 1, from, temp, to);

printf("Move disk %d from rod %c to rod %c.\n", num, from, to);

towerh(num - 1, temp, to, from);

}

int gcd(int a, int b)

{

if (a == 0)

return b;

if (b == 0)

return a;

if (a == b)

return a;

if (a > b)

return gcd(a - b, b);

return gcd(a, b - a);

}

void swap(char \*x, char \*y)

{

char temp;

temp = \*x;

\*x = \*y;

\*y = temp;

}

void revstr(char \*str, int num)

{

static int i = 0;

if (\*(str + num) == '\0')

return;

revstr(str, num + 1);

if (i <= num)

swap(&str[i++], &str[num]);

}

typedef struct node node;

struct node

{

int info;

node \*link;

};

struct node \*

create\_linked\_list(struct node \*start)

{

struct node \*temp, \*p;

int n;

printf("Enter the number of nodes: ");

scanf("%d", &n);

start = NULL;

if (n == 0)

return start;

for (int i = 0; i < n; i++)

{

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

printf("\nEnter the data for node %d: ", i + 1);

scanf("%d", &temp->info);

temp->link = NULL;

if (start == NULL)

start = temp;

else

{

p = start;

while (p->link != NULL)

p = p->link;

p->link = temp;

}

}

return start;

};

void search\_linked\_list(struct node \*start, int num)

{

static int i = 0;

i++;

if (start == NULL)

{

printf("Element not found.\n");

return;

}

else if (start->info == num)

{

printf("Element found in list at position %d.\n", i);

return;

}

else

{

search\_linked\_list(start->link, num);

}

}

int main()

{

int choice;

int base\_n, base\_c, num, a, b;

char str[20];

struct node \*start = NULL;

while (1)

{

printf("\nEnter 1 for base conversion.\n");

printf("Enter 2 for tower of hanoi.\n");

printf("Enter 3 for greatest common divisor.\n");

printf("Enter 4 to reverse a string.\n");

printf("Enter 5 to search an item in a linked list.\n");

printf("Enter 6 to exit.\n");

scanf("%d", &choice);

switch (choice)

{

case 1:

printf("Enter the base of the number entered and the base it is to be converted.\n");

scanf("%d%d", &base\_n, &base\_c);

printf("Enter the number to be converted.\n");

scanf("%d", &num);

base\_conv(num, base\_n, base\_c);

break;

case 2:

printf("Enter the number of disks.\n");

scanf("%d", &num);

printf("The moves involved in the Tower of Hanoi are :\n");

towerh(num, 'A', 'B', 'C');

break;

case 3:

printf("Enter the numbers.\n");

scanf("%d%d", &a, &b);

num = gcd(a, b);

printf("The GCD of the two numbers entered is %d.\n", num);

break;

case 4:

printf("Enter the string to be reversed.\n");

getchar();

fgets(str, 20, stdin);

revstr(str, 0);

printf("The reversed string is:\n");

fputs(str, stdout);

break;

case 5:

printf("Creating a linked list.\n");

start = create\_linked\_list(start);

printf("Enter the item to be searched.\n");

scanf("%d", &num);

search\_linked\_list(start, num);

break;

case 6:

exit(1);

default:

printf("Erroneous input.\n");

}

}

return 0;

}

**OUTPUT:-**





