NAME:	Shubham Vishwakarma
UID No.	2021700071
BRANCH:	S.Y CSE-DS
BATCH:	D
SUBJECT	Design and Analysis of Algorithms
EXPERIMENT No.	10
Date of Performance	16/04/2023
Date of Submission	20/04/2023

AIM:

String Matching algorithms (To implement Robin Karp algorithm)

Program 1

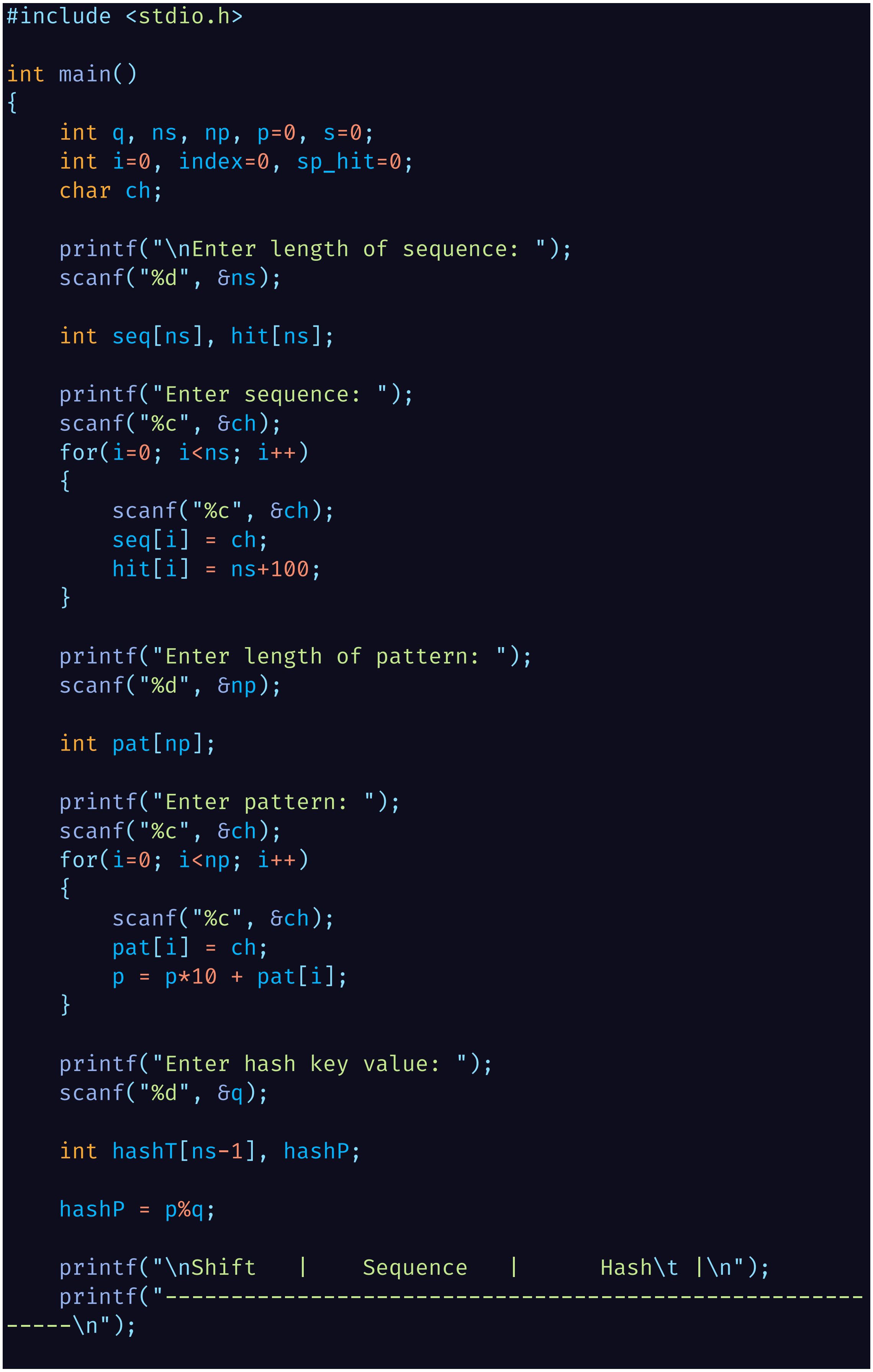
PROBLEM STATEMENT :

Implement the Robin Karp algorithm to the given sequence.

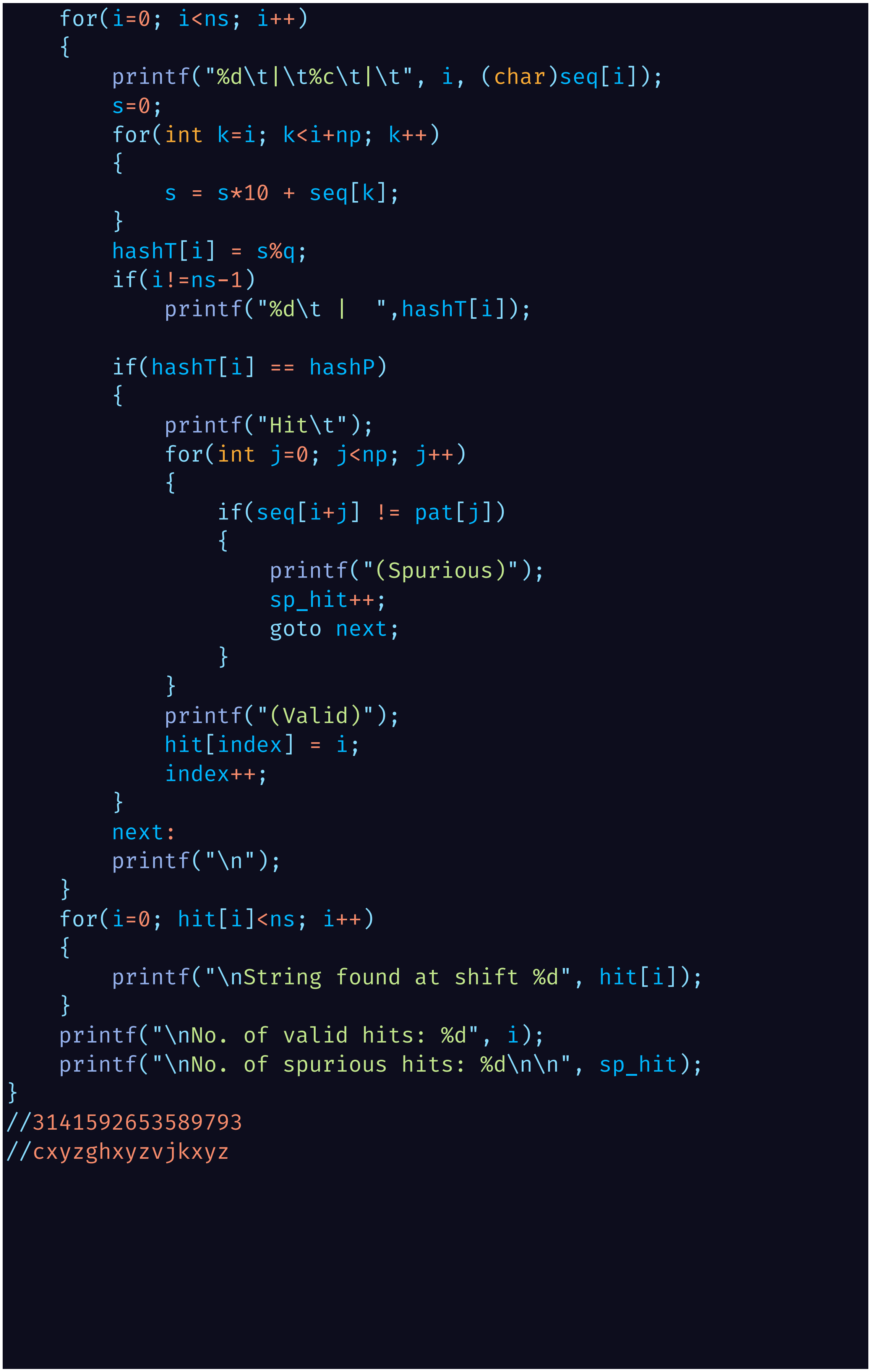
ALGORITHM/ THEORY:

```
Start
   pat_len := pattern Length
   str_len := string Length
   patHash := 0 and strHash := 0, h := 1
   maxChar := total number of characters in character set
for index i of all character in the pattern, do
   h := (h*maxChar) mod prime
for all character index i of pattern, do
   patHash := (maxChar*patHash + pattern[i]) mod prime
   strHash := (maxChar*strHash + text[i]) mod prime
for i := 0 to (str_len - pat_len), do
   if patHash = strHash, then
      for charIndex := 0 to pat_len -1, do
         if text[i+charIndex] ≠ pattern[charIndex], then
         break
if charIndex = pat_len, then
   print the location i as pattern found at i position.
if i < (str_len - pat_len), then
   strHash := (maxChar*(strHash – text[i]*h)+text[i+patLen]) mod prime,
then
   if strHash < 0, then
   strHash := strHash + primeEnd
```

PROGRAM:

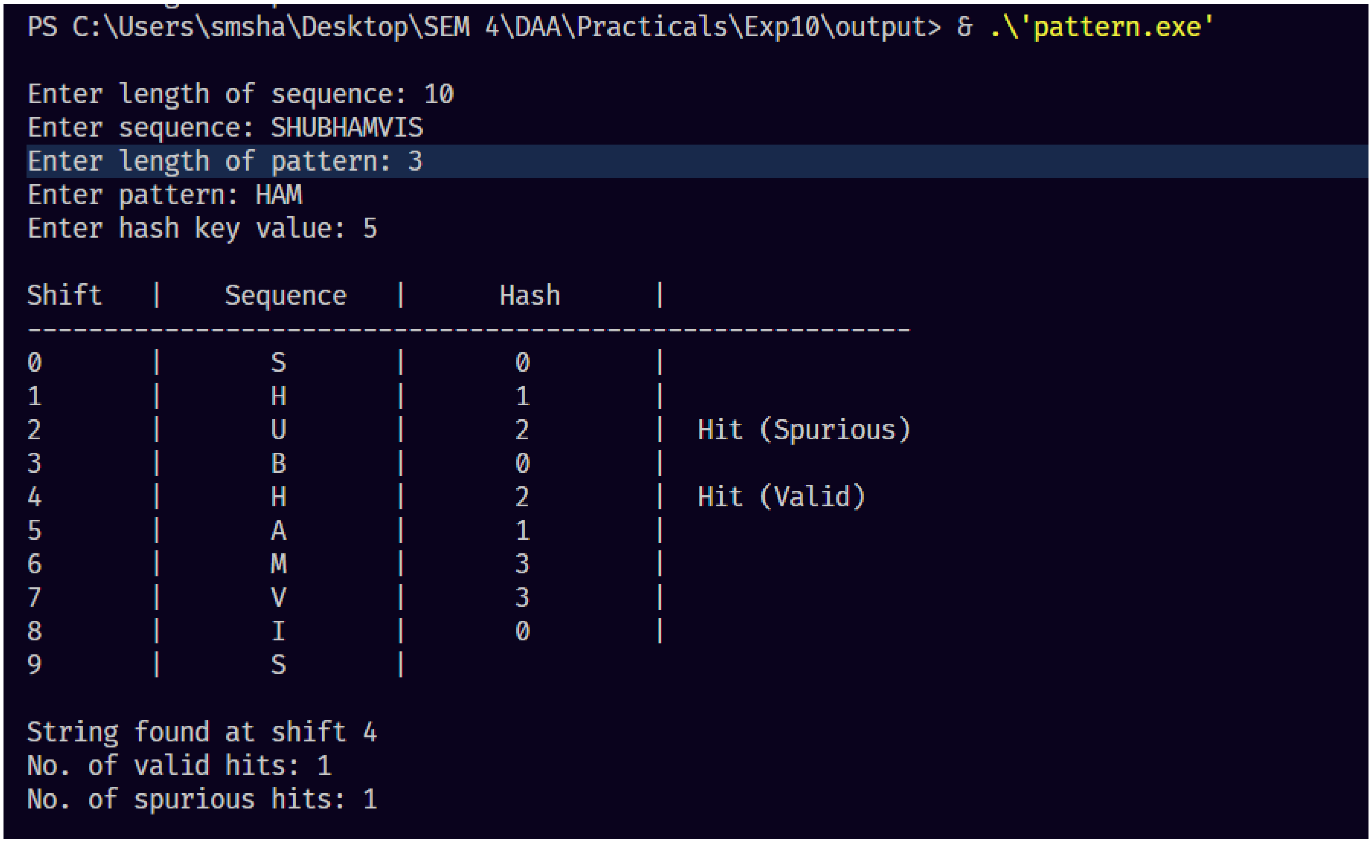


```c
#include <stdio.h>

int main()
{
    int q, ns, np, p=0, s=0;
    int i=0, index=0, sp_hit=0;
    char ch;

    printf("\nEnter length of sequence: ");
    scanf("%d", &ns);

    int seq[ns], hit[ns];

    printf("Enter sequence: ");
    scanf("%c", &ch);
    for(i=0; i<ns; i++)
    {
        scanf("%c", &ch);
        seq[i] = ch;
        hit[i] = ns+100;
    }

    printf("Enter length of pattern: ");
    scanf("%d", &np);

    int pat[np];

    printf("Enter pattern: ");
    scanf("%c", &ch);
    for(i=0; i<np; i++)
    {
        scanf("%c", &ch);
        pat[i] = ch;
        p = p*10 + pat[i];
    }

    printf("Enter hash key value: ");
    scanf("%d", &q);

    int hashT[ns-1], hashP;

    hashP = p%q;

    printf("\nShift   |    Sequence   |      Hash\t |\n");
    printf("-----------------------------------------------------
-----\n");
```



```
    for(i=0; i<ns; i++)
    {
        printf("%d\t|\t%c\t|\t", i, (char)seq[i]);
        s=0;
        for(int k=i; k<i+np; k++)
        {
            s = s*10 + seq[k];
        }
        hashT[i] = s%q;
        if(i!=ns-1)
            printf("%d\t |  ",hashT[i]);

        if(hashT[i] == hashP)
        {
            printf("Hit\t");
            for(int j=0; j<np; j++)
            {
                if(seq[i+j] != pat[j])
                {
                    printf("(Spurious)");
                    sp_hit++;
                    goto next;
                }
            }
            printf("(Valid)");
            hit[index] = i;
            index++;
        }
        next:
        printf("\n");
    }
    for(i=0; hit[i]<ns; i++)
    {
        printf("\nString found at shift %d", hit[i]);
    }
    printf("\nNo. of valid hits: %d", i);
    printf("\nNo. of spurious hits: %d\n\n", sp_hit);
}
//3141592653589793
//cxyzghxyzvjkxyz
```

RESULT:



```
PS C:\Users\smsha\Desktop\SEM 4\DAA\Practicals\Exp10\output> & .\'pattern.exe'

Enter length of sequence: 10
Enter sequence: SHUBHAMVIS
Enter length of pattern: 3
Enter pattern: HAM
Enter hash key value: 5

Shift   |    Sequence   |      Hash      |
----------------------------------------------------------
0       |       S       |       0        |
1       |       H       |       1        |
2       |       U       |       2        |  Hit (Spurious)
3       |       B       |       0        |
4       |       H       |       2        |  Hit (Valid)
5       |       A       |       1        |
6       |       M       |       3        |
7       |       V       |       3        |
8       |       I       |       0        |
9       |       S       |

String found at shift 4
No. of valid hits: 1
No. of spurious hits: 1
```

CONCLUSION :

Rabin-Karp algorithm **can be used to check image similarity based on hash value calculation**. The way the calculation works is the same as that done to string matching. Hence we achieved the aim of the experiment.