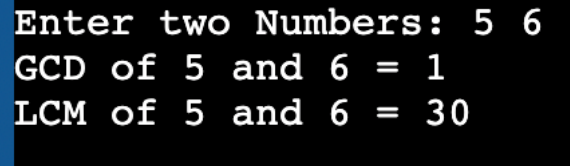


|                    |  |
|--------------------|--|
| AIM:               | Apply various control structures to solve given problems.  |
| Program 1          |  |
| PROBLEM STATEMENT: | Take two numbers as input and calculate their LCM and GCD (HCF)  |
| ALGORITHM:         | <p>Step 1: Read n1, n2</p> <p>Step 2: If n1&gt;n2:<br/>    numerator=n1<br/>    denominator=n2<br/>else:<br/>    numerator=n2<br/>    denominator=n1</p> <p>Step 3: Remainder=numerator%denominator</p> <p>Step 4: While Remainder !=0:<br/>    numerator=denominator<br/>    denominator=remainder<br/>    remainder=numerator% denominator<br/>end while</p> <p>Step 5: gcd=denominator<br/>    lcm=n1*n2/gcd</p> <p>Step 6: Print lcm and gcd</p>   |
| FLOWCHART:         | <pre>graph TD     Start([Start]) --&gt; Read[/Read n1, n2/]     Read --&gt; If{If n1 &gt; n2?}     If -- Yes --&gt; Set1[numerator=n1&lt;br/&gt;denominator=n2]     If -- No --&gt; Set2[numerator=n2&lt;br/&gt;denominator=n1]     Set1 --&gt; Remainder[remainder=numerator%denominator]     Set2 --&gt; Remainder     Remainder --&gt; While{while remainder!=0?}     While -- Yes --&gt; Update[numerator=denominator&lt;br/&gt;denominator=remainder&lt;br/&gt;remainder=numerator%denominator]     Update --&gt; While     While -- No --&gt; Calc[gcd=denominator&lt;br/&gt;lcm=n1*n2/gcd]     Calc --&gt; Print[/print lcm and gcd/]     Print --&gt; Stop([Stop])</pre> |

|  |  |
|--|--|
| <b>PROGRAM:</b>  | <pre> #include&lt;stdio.h&gt; int main() {     int n1,n2;     int gcd, lcm, remainder, numerator, denominator;      printf("Enter two Numbers: ");     scanf("%d %d",&amp;n1,&amp;n2);     if (n1&gt;n2)     {         numerator=n1;         denominator=n2;     }     else     {         numerator=n2;         denominator=n1;     }     remainder=numerator% denominator;     while(remainder!=0)     {         numerator=denominator;         denominator=remainder;         remainder=numerator% denominator;     }     gcd = denominator;     lcm = n1*n2/gcd;     printf("GCD of %d and %d = %d\n",n1,n2,gcd);     printf("LCM of %d and %d = %d\n",n1,n2,lcm);     return 0; } </pre> |
| <b>RESULT:</b><br> |  |
| <b>Program 2</b>   |  |
| <b>PROBLEM STATEMENT:</b>  | Write a program to convert a decimal number to binary or convert a binary number to decimal  |
| <b>ALGORITHM:</b>  | <p>Step 1: Divide the number by 2 through % (modulus operator) and store the remainder in array</p> <p>Step 2: Divide the number by 2 through / (division operator)</p> <p>Step 3: Repeat the step 2 until number is greater than 0</p>  |

|   |   |
|---|---|
| <b>FLOWCHART:</b>   | <pre> graph TD     Start([Start]) --&gt; Read[/Read the binary number from the user: n/]     Read --&gt; Init[Initialize d=0, i=0]     Init --&gt; LoopStart(( ))     LoopStart --&gt; Remainder[remainder = n % 10]     Remainder --&gt; Div[n = n / 10]     Div --&gt; Calc[d = d + (remainder * 2^i)]     Calc --&gt; Decision{Is n = 0?}     Decision -- Yes --&gt; Display[/Display d/]     Display --&gt; Stop([Stop])     Decision -- No --&gt; LoopStart   </pre> <p><small>Powered with free version of Visio. Full version doesn't put this mark.</small></p> |
| <b>PROGRAM:</b>   | <pre> #include &lt;stdio.h&gt; #include&lt;math.h&gt; int main() {     int binary,decimal,rem;     scanf("%d",&amp;decimal);     binary=0;     while (decimal&gt;0){         rem=decimal%2;         binary=(binary*10)+rem;         decimal/=2;     }     printf("%d",binary);     return 0; } </pre>   |
| <b>RESULT:</b><br><div style="background-color: black; color: white; padding: 10px; display: flex; justify-content: space-between;"> <span>Enter a Decimal Number: 5</span> <span>Decimal Number of 5 to Binary Number is 101</span> </div> |   |
| <b>Program 3</b>  |   |
| <b>PROBLEM STATEMENT:</b>   | Twin primes are consecutive odd numbers, both of which are prime numbers. Write a program which inputs two positive integers A and B and outputs all twin primes in range A to B.   |
| <b>ALGORITHM:</b>   | Step 1: Read start and end<br>Step 2: For i=start,i<end,i++:  |

|                   |  |
|-------------------|--|
|                   | <p>Step 1: If check_prime(i) and check_prime(i+2):<br/> Print {i,i+2} → consecutive prime number with difference of 2<br/> End if<br/> End for loop</p> <p><b>Function check_prime(num)</b><br/> Step 1: Read num from parameter<br/> Step 2: If num=1:<br/> return 0<br/> Step 3: for i=2, i&lt;num, i++:<br/> 1) if num%i=0:<br/> return 0<br/> end for loop<br/> Step 4: return 1</p>   |
| <b>FLOWCHART:</b> | <pre> graph TD     Start([Start]) --&gt; Input[/Input start,end/]     Input --&gt; Loop1{for(i = start, i &lt; end, i++)}     Loop1 --&gt; CheckPrime{if check_prime and check_prime(i+2)}     CheckPrime -- Yes --&gt; Print[/Print {i,i+2}/]     Print --&gt; Loop1     CheckPrime -- No --&gt; Stop([Stop])      subgraph Function_check_prime [Function check_prime]         direction TB         Start2([check_prime(num)]) --&gt; If1{If num=1}         If1 -- Yes --&gt; Return0_1[/Return 0/]         If1 -- No --&gt; Loop2{for(i = 2, i &lt; n, i++)}         Loop2 --&gt; If2{If num%i=0}         If2 -- Yes --&gt; Return0_2[/Return 0/]         If2 -- No --&gt; Return1[/Return 1/]     end </pre> |
| <b>PROGRAM:</b>   | <pre> #include&lt;stdio.h&gt; int check_prime(int n); int main(void) {     int start, end;     printf("Enter start: ");     scanf("%d", &amp;start);      printf("Enter end: ");     scanf("%d", &amp;end);      for(int i = start; i &lt; end; i++) </pre>  |

```

    {
        if(check_prime(i) && check_prime(i + 2))
        {
            printf("{%d, %d}\n", i, i + 2);
        }
    }
    return 0;
}
int check_prime(int n)
{
    if(n == 1)
    {
        return 0;
    }
    for(int i = 2; i < n; i++)
    {
        if(n % i == 0)
        {
            return 0;
        }
    }
    return 1;
}

```

#### RESULT:

```

Enter start: 3
Enter end: 20
{3, 5}
{5, 7}
{11, 13}
{17, 19}

```

#### Program 4

##### PROBLEM STATEMENT:

Write a program to find out whether a number is kaprekar or not. Consider an n-digit number k. Square it and add the right n digits to the left n or n-1 digits. If the resultant sum is k, then k is called a Kaprekar number. For example, 9 is a Kaprekar number since  $9^2=81$  and  $8+1=9$  and 297 is a Kaprekar number since  $297^2=88209$  and  $88+209=297$ . The first few are 1, 9, 45, 55, 99, 297, and 703.

##### ALGORITHM:

Step 1: Read num  
 Step 2: square\_num=num\*num  
         cout\_digit=0  
         sum=0  
 Step 3: While square\_num!=0:  
         cout\_digit = cout\_digit+1  
         square\_num= square\_num/10  
         end while  
 Step 4: square\_num=num\*num

|                          |  |
|--------------------------|--|
|                          | <p>Step 5: for r_digit=1, r_digit&lt;cout_digit, r_digit++:</p> <pre> eq_parts=10<sup>r_digit</sup> if eq_parts=num:     continue else:     sum = square_num / eq_parts + square_num % eq_parts     if sum=num:         Print num is Kaprekar number     else:         continue end for loop </pre> <p>Step 6: if sum!=num:<br/> Print num is not Kaprekar number<br/> End if</p> <p>Step 7: Stop</p>  |
| <p><b>FLOWCHART:</b></p> | <pre> graph TD     Start([Start]) --&gt; Read[/Read num/]     Read --&gt; Init[square_num=num*num&lt;br/&gt;cout_digit=0&lt;br/&gt;sum=0]     Init --&gt; While{while square_num&gt;0?}     While --&gt; CalcDigit[cout_digit=cout_digit+1&lt;br/&gt;square_num=square_num/10]     CalcDigit --&gt; CalcSquare[square_num=num*num]     CalcSquare --&gt; For{for (r_digit=1, r_digit&lt;cout_digit, r_digit++)}     For --&gt; CalcEqPart[eq_parts=10^r_digit]     CalcEqPart --&gt; IfEqPart{if eq_parts=num}     IfEqPart -- Yes --&gt; For     IfEqPart -- No --&gt; CalcSum[sum = square_num / eq_parts + square_num % eq_parts]     CalcSum --&gt; IfSumEq{if sum=num}     IfSumEq -- Yes --&gt; PrintYes[/Print num is Kaprekar number/]     IfSumEq -- No --&gt; IfSumNotEq{if sum!=num}     IfSumNotEq -- Yes --&gt; PrintNo[/Print num is not Kaprekar number/]     PrintYes --&gt; Stop([Stop])     PrintNo --&gt; Stop </pre> |
| <p><b>PROGRAM:</b></p>   | <pre> #include &lt;stdio.h&gt; #include&lt;math.h&gt; int main() {     int number,square_num,sum=0;     printf("Enter a number:");     scanf("%d",&amp;number);     square_num=number*number;     int cout_digit=0; </pre>   |

|  |   |
|--|---|
|  | <pre> while (square_num!=0){     cout_digit++;     square_num/=10; } square_num=number*number; for (int r_digit=1;r_digit&lt;cout_digit;r_digit++){     int eq_parts = pow(10, r_digit);      if (eq_parts == number)         continue;     sum = square_num / eq_parts + square_num % eq_parts;     if (sum == number){         printf("%d is a Kaprekar number",number);         break;     } } if (sum!=number)     printf("%d is not a Kaprekar number",number); return 0; } </pre> |
|--|---|

**RESULT:**

```

45
45 is a Kaprekar number

```

| Program 5                        |  |
|----------------------------------|--|
| <p><b>PROBLEM STATEMENT:</b></p> | <p>Write a program to print the following pattern</p> <p><b>Input:</b> 5</p> <p><b>Output:</b></p> <pre> ***** ***** *** ** * * ** *** ***** ***** </pre>  |
| <p><b>ALGORITHM:</b></p>         | <p>Step 1: Read num</p> <p>Step 2: for i=0, i&lt;num, i++:</p> <ol style="list-style-type: none"> <li>for j=0, j&lt;num, j++:             <ul style="list-style-type: none"> <li>print "*" → * from left</li> </ul> </li> <li>for j=0, j&lt;2*i+1, j++:             <ul style="list-style-type: none"> <li>print " " → space in between</li> </ul> </li> </ol> |

|                   |  |
|-------------------|--|
|                   | <pre> 3) for j=0, j&lt;num,j++:     print "*"→* from right end for loop end for loop Step 3: for i=0, i&lt;num-1, i++:     1) for j=0, j&lt;i+2, j++:         print "*"→* from left     end for loop     2) for j=0, j&lt;2*(n-i-1)+1,j++:         print " " →space in between     end for loop     3) for j=0, j&lt;i+2,j++:         print "*"→* from right     end for loop end for loop Step 4: Stop </pre>   |
| <b>FLOWCHART:</b> | <pre> graph TD     Start([Start]) --&gt; Input[/Input num/]     Input --&gt; Loop1{for (i = 0, i &lt; num, i++)}     Loop1 --&gt; Loop1a{for (j = 0, j &lt; num, j++)}     Loop1a --&gt; Print1[/Print */]     Print1 --&gt; Loop1b{for (j = 0, j &lt; 2*(n-i-1)+1, j++)}     Loop1b --&gt; Print2[/Print " /]     Print2 --&gt; Loop1c{for (j = 0, j &lt; num, j++)}     Loop1c --&gt; Print3[/Print */]     Print3 --&gt; Loop2{for (i = 0, i &lt; num-1, i++)}     Loop2 --&gt; Loop2a{for (j = 0, j &lt; i+2, j++)}     Loop2a --&gt; Print4[/Print */]     Print4 --&gt; Loop2b{for (j = 0, j &lt; 2*(n-i-1)+1, j++)}     Loop2b --&gt; Print5[/Print " /]     Print5 --&gt; Loop2c{for (j = 0, j &lt; i+2, j++)}     Loop2c --&gt; Print6[/Print */]     Print6 --&gt; Stop([Stop]) </pre> |
| <b>PROGRAM:</b>   | <pre> #include &lt;stdio.h&gt;  int main() </pre>  |



```

{
    int n;
    printf("Enter the number: ");
    scanf("%d",&n);
    for (int i = 0; i < n; i++) {
        for (int j = i; j < n; j++)
            printf("*");
        for (int j = 0; j < 2 * i + 1; j++)
            printf(" ");
        for (int j = i; j < n; j++)
            printf("*");
        printf("\n");
    }
    for (int i=0; i<n-1;i++){
        for(int j=0;j<i+2;j++)
            printf("*");
        for(int j=0;j<2*(n-i-1)-1;j++)
            printf(" ");
        for(int j=0;j<i+2;j++)
            printf("*");
        printf("\n");
    }
    return 0;
}

```

#### RESULT:

```

Enter the number: 5
*****  *****
*****   *****
***      ***
**         **
*          *
**         **
***      ***
*****   *****
*****  *****

```

#### CONCLUSION:

We learned to apply various control structures to solve given problems.