|  |  |
| --- | --- |
| **Name** | **Hatim Yusuf Sawai** |
| **UID no.** | **2021300108** |
| **Experiment No.** | **2** |

|  |  |
| --- | --- |
| **AIM:** | Apply various control structures to solve given problems. |
| **Program 1** | |
| **PROBLEM STATEMENT:** | Write a program to convert a decimal number to binary |
| **ALGORITHM:** | 1. START 2. Input dec 3. Initialize c = 1 4. While dec>0 repeatedly do   rem = dec % 2  dec = dec / 2  bin = bin + (rem \* c)  c \*= 10   1. print bin 2. STOP |
| **FLOWCHART:** |  |
| **PROGRAM:** | #include<stdio.h>  int main()  {      int dec,bin,rem,c=1;      printf("Enter a decimal number\n");      scanf("%d", &dec);      while (dec>0)      {          rem = dec % 2;          dec = dec / 2;          bin = bin + (rem \* c);          c \*= 10;      }      printf("Binary equivalent = %d\n", bin);      return 0;  } |
| **RESULT:** | |
| **Program 2** | |
| **PROBLEM STATEMENT:** | 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. |
| **ALGORITHM:** | 1. START 2. Input a,b 3. Initialize check=0 4. If a is even   i = a+1   1. Else   i = a+2   1. While i<=b repeatedly do   If a and i are odd  If a and i are (prime)  Print a,i  Check=1  a = i  i = i + 2   1. If check == 0   Print no pairs found   1. STOP |
| **PROGRAM:** | #include<stdio.h>  #include<math.h>  int prime(int);  int main()  {      int a,b,i,check=0;      printf("Enter the range a to b:\n");      scanf("%d%d",&a,&b);      for(i=a;i+2<=b;i++)      {          if (i%2!=0 && (i+2)%2!=0)          {              if (prime(i) == 0 && prime(i+2) == 0)              {                  printf("(%d, %d) ", i, i+2);                  check = 1;              }          }      }      if(check==0)          printf("No prime pairs found");      return 0;  }  int prime(int n)  {      if (n == 1)          return 1;      else if (n > 1)      {          for (int i = 2; i <= sqrt(n); i++)          {              if (n % i == 0)                  return 1;          }      }      return 0;  } |
| **RESULT:** | |
| **Program 3** | |
| **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 |
| **ALGORITHM:** | 1. START 2. Input n 3. sq = n\*n 4. For i=0   sq = sq/10  c++   1. Repeat step 4 till sq>0 2. sq = n\*n 3. for i=1   k = 10^i  sum = sq/k + sq%k  if sum==n  output kaprekar number  flag =1   1. repeat step 7 till i<=c 2. if flag==0   output not a kaprekar number   1. STOP |
| **PROGRAM:** | #include<stdio.h>  #include<math.h>  int main()  {      int n,sq,sum,i,c=0,k,flag=0;      printf("Enter a number:\n");      scanf("%d", &n);      sq=n\*n;      for(i=0;sq>0;i++)      {          sq=sq/10;          c++;      }      sq = n\*n;      for(i=1;i<=c;i++)      {          k = pow(10,i);          sum = sq/k + sq%k;          if(sum==n)          {              printf("%d is a kaprekar number",n);              flag=1;              break;          }      }      if(flag==0)          printf("%d is not a kaprekar number",n);      return 0;  } |
| **RESULT:** | |
| **Program 4** | |
| **PROBLEM STATEMENT:** | Note that 12\*42 = 21\*24 and 12\*63 = 21\*36 and 12\*84 = 21\*48 and so on. There is a property that (10a+b)\*(10c+d) =(10b+a)(10d+c) where a and b are unequal and c and d are also unequal. Write a program which outputs them all between 10 to 99. |
| **ALGORITHM:** | 1. START 2. a = 10,b = 99 3. for a=10   for b = 10   1. c = (10 \* (a % 10)) + a / 10 2. d = (10 \* (b % 10)) + b / 10 3. if ((a\*b == c\*d) && (c!=a) && (b!=d) && (c!=b) && (d!=a))   Output a\*b = c\*d  Repeat steps 1-3 till b<100  4. Repeat step 3 till a<100  5. STOP |
| **FLOWCHART:** |  |
| **PROGRAM:** | #include <stdio.h>  int main()  {      int a, b, c, d;      for(a=10;a<100;a++)      {          for(b=10;b<100;b++)          {              c = (10 \* (a % 10)) + a / 10;              d = (10 \* (b % 10)) + b / 10;              if ((a\*b == c\*d)&&(c!=a)&&(b!=d)&&(c!=b)&&(d!=a))              {                  printf("%d\*%d = %d\*%d\n", a, b, c, d);              }          }      }  } |
| **RESULT:** | |
| **Program 5** | |
| **PROBLEM STATEMENT:** | Take two numbers as input and calculate their LCM and GCD (HCF). |
| **ALGORITHM:** | 1. START 2. Input a,b 3. For i=1   If a%i==0 and b%i==0  Gcd = i   1. Repeat step 3 till i<=a 2. lcm = (a\*b)/gcd 3. Ouput lcm,gcd 4. STOP |
| **PROGRAM:** | #include<stdio.h>  int main()  {      int a,b,gcd,lcm,i;      printf("Enter the two numbers:\n");      scanf("%d%d",&a,&b);        for(i=1;i<=a;i++)      {          if(a%i==0&&b%i==0)          {              gcd=i;          }      }      lcm = (a\*b)/gcd;      printf("GCD = %d\nLCM = %d\n",gcd,lcm);      return 0;  } |
| **RESULT:** | |
| **CONCLUSION:** | In this experiment, we learnt how to use various control flow statements like: while loop, for loop and nested loops in our programs |