|  |  |
| --- | --- |
| **Name** | Adwait S Purao |
| **UID no.** | 2021300101 |
| **Experiment No.** | 2 |

|  |  |
| --- | --- |
| **AIM:** | Apply various control structures to solve given problems. |
| **Program 1** | |
| **PROBLEM STATEMENT :** | Take two numbers as input and calculate their LCM and GCD |
| **ALGORITHM:** | 1. START  2. Input 2 Numbers  3. if (n1 > n2)  {  num = n1  den = n2  }  else  {  num = n2  den = n1  }  4. while (rem != 0)  {  num = den  den = rem  rem = num % den  }  5. gcd = den  lcm = n1 \* n2 / gcd  6.Print LCM and GCD  7.STOP |
| **FLOWCHART:** |  |
| **PROGRAM:** | #include <stdio.h>    int main()  {  int n1, n2, gcd, lcm, rem, num, den;    printf("Enter two numbers\n");  scanf("%d %d", &n1, &n2);  if (n1 > n2)  {  num = n1;  den = n2;  }  else  {  num = n2;  den = n1;  }  rem = num % den;  while (rem != 0)  {  num = den;  den = rem;  rem = num % den;  }  gcd = den;  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;  } |
| **RESULT:** | |
| **Program 2** | |
| **PROBLEM STATEMENT :** | Write a program to convert a decimal number to binary or convert a binary  number to decimal |
| **ALGORITHM:** | 1. START  2.Input 1 for decimal and two for binary to decimal  3. If Inp =1  Input Decimal Number  while(n!=0)  {  bi = bi + ((n % 2)\* i);  n = n / 2;  i = i \* 10;  }  4.print bi  5. Else If Input = 2  Input Binary Number  Sum = 0  For(I=0;Number>0;I++)  {  Temp = (Number-((Number/10)\*10))\*(2^I)  Sum = Sum + Temp  }  Print Sum  6. STOP |
| **FLOWCHART:** |  |
| **PROGRAM:** | #include <stdio.h>  #include<math.h>  int main()  {  int flag=0;  int inp;  do  {  printf("If you want to convert Decimal number to Binary number, type 1 and if you want to convert Binary number to Decimal number, then type 2\n");  scanf("%d",&inp);  if(inp==1)  {  flag=1;  int n, bi=0, i=1;  printf("Enter a number: ");  scanf("%d", &n);  printf ("%d in binary is ", n);  while(n!=0)  {  bi = bi + ((n % 2)\* i);  n = n / 2;  i = i \* 10;  }  printf("%d", bi);    }  else if(inp==2)  {  flag=1;  int bi,n,temp,sum=0;  printf("Enter the Binary Number: \n");  scanf("%d",&bi);  n=bi;  for(int i=0;n>0;i++)  {  temp=(n-((n/10)\*10))\*pow(2,i);  n=n/10;  sum = sum + temp;  }  printf("Decimal of %d in Binary is %d\n",bi,sum);  }  else  {  flag=0;  printf("Wrong Choice\n");  }  }while(flag==0);  return 0;  } |
| **RESULT:** | |
| **Program 3** | |
| **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.Enter starting number and ending number  3. for(i=n1; i<=n2; i++)      {          for(j=i-1; j>1; j--)          {              if(i%j == 0)              {                  prmchk = 0;                  break;              }          }          if(prmchk == 1)   {              prime1 = i;              if(prime1 - prime2 == 2)  4.print prime1 and prime2  5.STOP |
| **FLOWCHART:** |  |
| **PROGRAM:** | #include <stdio.h>  int main()  {      int n1, n2, i, j;      int prime1, prime2, prmchk=1;      printf("Enter starting number: ");      scanf("%d", &n1);      printf("Enter ending number: ");      scanf("%d", &n2);      for(i=n1; i<=n2; i++)      {          for(j=i-1; j>1; j--)          {              if(i%j == 0)              {                  prmchk = 0;                  break;              }          }          if(prmchk == 1)   {              prime1 = i;              if(prime1 - prime2 == 2)              {                  printf("(%d,%d), ", prime1, prime2);              }          }          prime2 = prime1;          prmchk = 1;      }      return 0;  } |
| **RESULT:** | |
| **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.4 |
| **ALGORITHM:** | 1.START  2.input a number n  3.square=n\*n  4. while(sq != 0)      {          digits++;          sq = sq/10;      }      if(digits%2 == 0){}      else {          digits = digits + 1;      }  5. num = digits/2;      power = pow(10, num);      k1 = sq % power;      k2 = (sq - k1) / power;      if(k1+k2 == n)  Print it’s a Kaprekar number  Else  Print it’s not a kaprekar number  6.STOP |
| **FLOWCHART:** |  |
| **PROGRAM:** | #include <stdio.h>  #include <math.h>  int main()  {      int n, k1, k2, sq, digits, num, power;      printf("Enter a number: ");      scanf("%d", &n);      sq = n\*n;      while(sq != 0)      {          digits++;          sq = sq/10;      }      if(digits%2 == 0){}      else {          digits = digits + 1;      }      sq = n\*n;      num = digits/2;      power = pow(10, num);      k1 = sq % power;      k2 = (sq - k1) / power;      if(k1+k2 == n)      {          printf("It's a Kaprekar number");      }else {printf("It's not a Kaprekar number");}      return 0;  } |
| **RESULT:** | |
| **Program 5** | |
| **PROBLEM STATEMENT:** | Write a program to check whether a given number is Armstrong number or  not. For Example 371 is 3^3+7^3+1^3=371. |
| **ALGORITHM:** | 1.START  2.Take input number from user  store number in num and temp variable n  3. while(n!=0)      {          n = n/10;          digCount++;      }      n = num;      while(n!=0)      {          dig = n%10;          power = pow(dig, digCount);          sum = sum + power;          n = n/10;      }      if (sum == num)  4.Print num is an armstrong number  5.else  Print it’s not aan armstrong number  6.STOP |
| **FLOWCHART:** |  |
| **PROGRAM:** | #include <stdio.h>  #include <math.h>  int main()  {      int num, dig, power, sum=0, digCount=0, n;      printf("Enter a number: ");      scanf("%d", &num);      n = num;      while(n!=0)      {          n = n/10;          digCount++;      }      n = num;      while(n!=0)      {          dig = n%10;          power = pow(dig, digCount);          sum = sum + power;          n = n/10;      }      if (sum == num)      {          printf("%d is an armstrong number", num);      }else { printf("%d is not an armstrong number", num); }      return 0;  } |
| **RESULT:** | |
| **CONCLUSION:** | In this experiment we learnt about various types of control structures like for,while, do while we also learnt about the if-elsse statement and the various types of compund assignment operators like !=(not equals),==(equals to) and arithmetic operators like /,+,- etc. |