| **NAME:** | Darsh Shah |
| --- | --- |
| **UID No.** | 2021700057 |
| **BRANCH:** | SY CSE DS |
| **BATCH:** | D4 |
| **SUBJECT** | DAA |
| **EXPERIMENT No.** | 1 |
| **DATE:** | 6th February , 2023 |

| **AIM:** | Aim – To implement the various functions e.g. linear, non-linear, quadratic, exponential etc. |
| --- | --- |
| **Program 1** | |
| **PROBLEM STATEMENT :** | A function is a relation between a set of inputs and a set of permissible outputs with the property that each  input is related to exactly one output. Let A & B be any two non-empty sets; mapping from A to B will be a function  only when every element in set A has one end, only one image in set B. |
| GRAPHS: | Graph 1 is for functions:- 2,4,5,10,11  Graph 2 is for functions:- 3,6,7,8,9  Graph 3 specifically for factorial function |
| **PROGRAM:** | #include<stdio.h>  #include<math.h>  double func1(double i)  {  double r = 0.0;  r = pow(1.5 , i);  return r;  }  double func2(double i)  {  double r = 0.0;  r = pow(i , 3);  return r;  }  double func3(double i)  {  double r = 0.0;  r = i;  return r;  }  double func4(double i)  {  double r = 0.0;  r = pow(2 , i);  return r;  }  double func5(double i)  {  double e = 2.71;  double r = 0.0;  r = pow(e , i);  return r;  }  double func6(double i)  {  double r = 0.0;  r = log(i);  return r;  }  double func7(double i)  {  double r = 0.0;  r = log2(i);  return r;  }  double func8(double i)  {  double r = 0.0;  r = log(log(i));  return r;  }  double func9(double i)  {  double r = 0.0;  r = log2(log2(i));  return r;  }  double func10(double i)  {  double r = 0.0;  r = pow(2 , pow( 2 , i + 1));  return r;  }  double func11(double i)  {  if(i)  {  return ( i \* func11(i-1));  }  else  {  return 1;  }  }  int main()  {  int i=0;  double x=0;  printf("x = (3/2)^n \n");  for(i=0;i<=100;i=i+10) // x = (3/2)^n  {  x = func1(i);  printf("%d " , i);  printf("\t");  printf("%.2f \n" , x);  }  printf("\n");  printf("x = n^3 \n");  for(i=0;i<=100;i=i+10) // x = n^3  {  x = func2(i);  printf("%d " , i);  printf("\t");  printf("%.2f \n" , x);  }    printf("\n");  printf("x = n \n");  for(i=0;i<=100;i=i+10) // x = n  {  x = func3(i);  printf("%d " , i);  printf("\t");  printf("%.2f \n" , x);  }    printf("\n");  printf("x = 2^n \n");  for(i=0;i<=100;i=i+10) // x = 2^n  {  x = func4(i);  printf("%d " , i);  printf("\t");  printf("%.2f \n" , x);  }    printf("\n");  printf("x = e^n \n");  for(i=0;i<=100;i=i+10) // x = e^n  {  x = func5(i);  printf("%d " , i);  printf("\t");  printf("%.2f \n" , x);  }    printf("\n");  printf("x = loge n \n");  for(i=0;i<=100;i=i+10) // x = loge n  {  x = func6(i);  printf("%d " , i);  printf("\t");  printf("%.2f \n" , x);  }  printf("\n");  printf("x = log2 n \n");  for(i=0;i<=100;i=i+10) // x = log2 n  {  x = func7(i);  printf("%d " , i);  printf("\t");  printf("%.2f \n" , x);  }    printf("\n");  printf("x = loge loge n \n");  for(i=0;i<=100;i=i+10) // x = loge loge n  {  x = func8(i);  printf("%d " , i);  printf("\t");  printf("%.2f \n" , x);  }    printf("\n");  printf("x = log2 log2 n \n");  for(i=0;i<=100;i=i+10) // x = log2 log2 n  {  x = func9(i);  printf("%d " , i);  printf("\t");  printf("%.2f \n" , x);  }    printf("\n");  printf("x = 2 ^ 2 ^ (n + 1) \n");  for(i=0;i<=10;i=i+1) // x = 2 ^ 2 ^ (n + 1)  {  x = func10(i);  printf("%d " , i);  printf("\t");  printf("%.2f \n" , x);  }    printf("\n");  printf("x = n! \n");  for(i=0;i<=20;i=i+2) // x = 2 ^ 2 ^ (n + 1)  {  x = func11(i);  printf("%d " , i);  printf("\t");  printf("%.2f \n" , x);  }      } |
| **RESULT:** | x = (3/2)^n  0 1.00  10 57.67  20 3325.26  30 191751.06  40 11057332.32  50 637621500.21  60 36768468716.93  70 2120255184830.25  80 122264598055704.64  90 7050392822843069.00  100 406561177535215232.00  x = n^3  0 0.00  10 1000.00  20 8000.00  30 27000.00  40 64000.00  50 125000.00  60 216000.00  70 343000.00  80 512000.00  90 729000.00  100 1000000.00  x = n  0 0.00  10 10.00  20 20.00  30 30.00  40 40.00  50 50.00  60 60.00  70 70.00  80 80.00  90 90.00  100 100.00  x = 2^n  0 1.00  10 1024.00  20 1048576.00  30 1073741824.00  40 1099511627776.00  50 1125899906842624.00  60 1152921504606846976.00  70 1180591620717411303424.00  80 1208925819614629174706176.00  90 1237940039285380274899124224.00  100 1267650600228229401496703205376.00  x = e^n  0 1.00  10 21364.51  20 456442228.94  30 9751663938615.36  40 208339508359304896.00  50 4451071224011028955136.00  60 95094949571691216833085440.00  70 2031656870656286753797869404160.00  80 43405350743396877971235172998184960.00  90 927333990482707166514346952223881166848.00  100 19812035041219038986232623646666779910995968.00  x = loge n  0 -inf  10 2.30  20 3.00  30 3.40  40 3.69  50 3.91  60 4.09  70 4.25  80 4.38  90 4.50  100 4.61  x = log2 n  0 -inf  10 3.32  20 4.32  30 4.91  40 5.32  50 5.64  60 5.91  70 6.13  80 6.32  90 6.49  100 6.64  x = loge loge n  0 -nan  10 0.83  20 1.10  30 1.22  40 1.31  50 1.36  60 1.41  70 1.45  80 1.48  90 1.50  100 1.53  x = log2 log2 n  0 -nan  10 1.73  20 2.11  30 2.29  40 2.41  50 2.50  60 2.56  70 2.62  80 2.66  90 2.70  100 2.73  x = 2 ^ 2 ^ (n + 1)  0 4.00  1 16.00  2 256.00  3 65536.00  4 4294967296.00  5 18446744073709551616.00  6 340282366920938463463374607431768211456.00  7 115792089237316195423570985008687907853269984665640564039457584007913129639936.00  8 13407807929942597099574024998205846127479365820592393377723561443721764030073546976801874298166903427690031858186486050853753882811946569946433649006084096.00  9 inf  10 inf  x = n!  0 1.00  2 2.00  4 24.00  6 720.00  8 40320.00  10 3628800.00  12 479001600.00  14 87178291200.00  16 20922789888000.00  18 6402373705728000.00  20 2432902008176640000.00 |
| **CONCLUSION:** | Successfully studied various functions. |