PRACTICAL – 5

PROGRAM -1

AIM- Write a program in C to find the square of any number using the function

CODE:-

|  |
| --- |
| #include <stdio.h>  int square(int num) {  return num \* num;  }  int main() {  int number, sq;  printf("Enter a number to find its square: ");  scanf("%d", &number);  sq = square(number);  printf("Square of %d = %d\n", number, sq);  return 0;  } |

Output:-

|  |
| --- |
|  |

PROGRAM -2

AIM- Write a program in C to find the square of any number using the function

CODE:- OUTPUT:-

|  |  |
| --- | --- |
| |  | | --- | |  |   #include <stdio.h>  int isPrime(int num)  {  int i;  if (num <= 1)  {  return 0; // Not a prime number  }  for ( i = 2; i <= num / 2; i++) {  if (num % i == 0) {  return 0; // Not a prime number  }  }  return 1; // Prime number  }  int main()  {  printf("\n HARSH D \n");  int number, isPrimeNumber;  printf("Enter a number to check if it is a prime number: ");  scanf("%d", &number);  isPrimeNumber = isPrime(number);  if (isPrimeNumber == 1) {  printf("%d is a prime number.\n", number);  } else {  printf("%d is not a prime number.\n", number);  }  return 0;  } |

PROGRAM -3

AIM- Write a program in C to check whether two given strings are an anagram. Test Data : Input the first String : spare Input the second String : pears

CODE:-

|  |
| --- |
| #include <stdio.h>  #include <string.h>  void sortString(char \*str) {  int len = strlen(str);  int i, j, temp;  // Bubble sort to sort the characters of the string  for (i = 0; i < len-1; i++) {  for (j = i+1; j < len; j++) {  if (str[i] > str[j]) {  temp = str[i];  str[i] = str[j];  str[j] = temp;  }  }  }  }  int checkAnagram(char \*str1, char \*str2)  {  printf("\n HARSH D \n");  // Length of both strings  int len1 = strlen(str1);  int len2 = strlen(str2);  // If lengths are not equal, they can't be anagrams  if (len1 != len2) {  return 0;  }  // Sort both strings  sortString(str1);  sortString(str2);  // Compare the sorted strings  if (strcmp(str1, str2) == 0) {  return 1; // Anagram  } else {  return 0; // Not an Anagram  }  }  int main() {  char str1[100], str2[100];  printf("Input the first String: ");  fgets(str1, 100, stdin);  printf("Input the second String: ");  fgets(str2, 100, stdin);  // Remove newline characters from input strings  str1[strcspn(str1, "\n")] = 0;  str2[strcspn(str2, "\n")] = 0;  if (checkAnagram(str1, str2)) {  printf("The strings are anagrams.\n");  } else {  printf("The strings are not anagrams.\n");  }  return 0;  } |

OUTPUT:-

|  |
| --- |
|  |

PROGRAM -4

AIM- Write a C function to find whether the number entered is Armstrong or not.

CODE:-

|  |
| --- |
| #include <stdio.h>  #include <math.h>  int isArmstrong(int num) {  int originalNum, remainder, result = 0, n = 0;  originalNum = num;  // Find the number of digits  while (originalNum != 0) {  originalNum /= 10;  ++n;  }  originalNum = num;  // Check if the number is Armstrong  while (originalNum != 0) {  remainder = originalNum % 10;  result += pow(remainder, n);  originalNum /= 10;  }  if (result == num) {  return 1; // Armstrong number  } else {  return 0; // Not an Armstrong number  }  }  int main() {  int num;  printf("Enter a number to check for Armstrong: ");  scanf("%d", &num);  if (isArmstrong(num)) {  printf("%d is an Armstrong number.\n", num);  } else {  printf("%d is not an Armstrong number.\n", num);  }  return 0;  } |

OUTPUT:-

|  |
| --- |
|  |

PROGRAM -5

AIM- Write a C Program to find Factorial of Given Number.

CODE:-

|  |
| --- |
| #include <stdio.h>  int factorial(int n)  {  if (n == 0) {  return 1;  } else {  return n \* factorial(n - 1);  }  }  int main()  {  printf("\n HARSH D \n");  int num;  printf("Enter a number: ");  scanf("%d", &num);  if (num < 0) {  printf("Factorial of a negative number is undefined\n");  } else {  int result = factorial(num);  printf("Factorial of %d is %d\n", num, result);  }  return 0;  } |

OUTPUT:-

|  |
| --- |
|  |

PROGRAM -5

AIM- Write a C Program to find Fibonacci of Given Number.

CODE:-

|  |
| --- |
| #include <stdio.h>  int fibonacci(int n)  {  if (n <= 1) {  return n;  } else {  return fibonacci(n - 1) + fibonacci(n - 2);  }  }  void printFibonacci(int count){  int i;  for (i = 0; i < count; i++) {  printf("%d ", fibonacci(i));  }  printf("\n");  }  int main()  {  printf("\n HARSH D \n");  int count;  printf("Enter the number of terms in the Fibonacci series: ");  scanf("%d", &count);  printf("Fibonacci Series up to %d terms:\n", count);  printFibonacci(count);  return 0;  } |

OUTPUT:-

|  |
| --- |
|  |