1. Write a C Program to find HCF of 4 given numbers using recursive function

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Private Test cases used for evaluation** | **Input** | **Expected Output** | **Actual Output** | **Status** |
| Test Case 1 | 50  455  60  200 | The HCF is 5 | The HCF is 5 | Passed |
| Test Case 2 | 67  89  45  41 | The HCF is 1 | The HCF is 1 | Passed |

#include<stdio.h>

int HCF(int, int); //You have to write this function which calculates the HCF.

int main()

{

int a, b, c, d, result;

scanf("%d %d %d %d", &a, &b, &c, &d); /\* Takes 4 number as input from the test data \*/

result = HCF(HCF(a, b), HCF(c,d));

printf("The HCF is %d", result);

}

//Complete the rest of the program to calculate HCF

int HCF(int x,int y)

{

while (x!=y)

{

if(x>y)

{

return HCF(x-y,y);

}

else

{

return HCF(x,y-x);

}

return x;

}

}

1. Write a C Program to find power of a given number using recursion. The number and the power to be calculated is taken from test case

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Private Test cases used for evaluation** | **Input** | **Expected Output** | **Actual Output** | **Status** |
| Test Case 1 | 5  3 | 5^3 is 125 | 5^3 is 125 | Passed |
| Test Case 2 | 16  3 | 16^3 is 4096 | 16^3 is 4096 | Passed |

#include <stdio.h>

long power(int, int);

int main()

{

int pow, num;

long result;

scanf("%d", &num); //The number taken as input from test case data

scanf("%d", &pow); //The power is taken from the test case

result = power(num, pow);

printf("%d^%d is %ld", num, pow, result);

return 0;

}

long power(int num, int pow)

{

if(pow)

{

return(num\*power(num,pow-1));

}

return 1;

}

1. Write a C Program to print Binary Equivalent of an Integer using Recursion

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Private Test cases used for evaluation** | **Input** | **Expected Output** | **Actual Output** | **Status** |
| Test Case 1 | 30 | The binary equivalent of 30 is 11110 | The binary equivalent of 30 is 11110\n | Passed |
| Test Case 2 | 111 | The binary equivalent of 111 is 1101111 | The binary equivalent of 111 is 1101111\n | Passed |

#include <stdio.h>

int binary\_conversion(int); //function to convert binary to decimal number

int main()

{

int num, bin; //num is the decimal number and bin is the binary equivalent for the number

scanf("%d", &num); //The decimal number is taken from the test case data

bin = binary\_conversion(num); //binary number is stored in variable bin

printf("The binary equivalent of %d is %d\n", num, bin);

return 0;

}

int binary\_conversion(int num)

{

if(num==0)

{

return 0;

}

else

{

return (num%2)+10\* binary\_conversion(num/2);

}

}

1. Write a C Program to reverse a given word using function. e.g. INDIA should be printed as AIDNI

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Private Test cases used for evaluation** | **Input** | **Expected Output** | **Actual Output** | **Status** |
| Test Case 1 | INDIA | The string after reversing is: AIDNI | The string after reversing is: AIDNI | Passed |
| Test Case 2 | DELHI | The string after reversing is: IHLED | The string after reversing is: IHLED | Passed |

#include<stdio.h>

#include<string.h>

void reverse(char[], int, int);

int main()

{

char str1[20];

scanf("%s", str1); //The string is taken as input form the test data.

/\* Complete the program to print the string in reverse order using the function

void reverse(char[], int, int);

Use the printf statement as

printf("The string after reversing is: %s\n", str1);

You can use different variable also.

\*/

reverse(str1, strlen(str1), 0);

printf("The string after reversing is: %s", str1);

return 0;

}

void reverse(char str1[], int len, int i)

{

char temp;

for(int i=0; i<len/2; i++)

{

temp = str1[i];

str1[i] = str1[len-i-1];

str1[len-i-1] = temp;

}

str1[len]='\0';

}