TCS Ninja coding questions and answers

TCS Ninja coding questions are a must to prepare to clear the TCS Ninja test. We are going to discuss all the previously asked TCS Ninja coding questions in this article. These will help you prepare for any kind of coding questions asked in the exam.

TCS Ninja Mock test questions – Coding section

Consider the below series:

1, 2, 1, 3, 2, 5, 3, 7, 5, 11, 8, 13, 13, 17, …

This series is a mixture of 2 series – all the odd terms in this series form a Fibonacci series and all the even terms are the prime numbers in ascending order.

Write a program to find the Nth term in this series.

The value N is a Positive integer that should be read from STDIN. The Nth term that is calculated by the program should be written to STDOUT. Other than the value of Nth term, no other characters/strings or message should be written to STDOUT.

For example, when N = 14, the 14th term in the series is 17. So only the value 17 should be printed to STDOUT.

Program:

```
#include<stdio.h>

int i, t1 = 0, t2 = 1, nextTerm;
for (i = 1; i<=n; i++)
{
nextTerm = t1 + t2;
t1 = t2;
t2 = nextTerm;
}
printf("%d", t1);
}

void prime(int n)
{
int i, j, flag, count =0;
for (i=2; i<=MAX; i++)
{
flag = 0;
for (j=2; j<i; j++)
{
if(i%j == 0)
{
flag = 1;
break;
}
}
if (flag == 0)
count++;
if(count == n)
{
printf("%d", i);
break;
}
```

```
}
```

```
int n;
scanf("%d", &n);
if(n%2 == 1)
fibonacci (n/2 + 1);
else
prime(n/2);
return 0;
}
```

TCS Ninja Coding question 1:

Factorial program in c using command line arguments.

Explanation: Factorial of a non-negative integer n, denoted by n!, is the product of all positive integers less than or equal to n. For example, The value of 5! is 5*4*3*2*1 = 120

Solution:

```
#include
int main(int a, char *b[])  //command line arguments
{
int x,y,f=1;
x=atoi(b[1]);  //atoi function is to convert a character to integer
for(i=1;i<=x;i++)
{
f=f*i;
}
printf("%d",f);
return 0;
}
```

TCS Ninja Coding question 2:

variable with 2 point precision.

Solution:

```
#include
#define PI 3.14
int main(int a, char *b[])   //command line arguments
{
int d; float area =0;
d= atoi(argv[1]);
area =(float) PI*(d/2)*(d/2);
printf("%0.2f", area);    //%0.2f is to print the answer with 2 values after decimal point.
return 0;
}
```

TCS Ninja Coding question 3:

Write a c program, to check whether the given year is a leap year or not using command line arguments. A leap year is a calendar year containing one additional day (Feb 29th) added to keep the calendar year synchronized with the astronomical year.

Solution:

```
#include
int main(int a, char*b[])
{
int year; year=atoi(b[1]);
if(year%100==0)
{
if(year%400==0)
```

```
{
```

```
printf(“NOT LEAP YEAR”); } }
else if(year%4==0)
{
printf(“LEAP YEAR”);
}
else{
printf(“NOT LEAP YEAR”);
}
return 0; }
```

TCS Ninja Coding question 4:

Write a c program, to find the GCD of the given 2 numbers, using command line arguments. The input is 2 integer and the output GCD also should be an integer value.

Solution:

```
#include
int main(int x, char *y[])
{
inta,b,small,i;
a=atoi(y[1]);
b=atoi(y[2]);
small=a>b?b:a;
for(i=small;i>=1;i–)
{
if((a%i==0)&&(b%i==0))
{
printf(“%d”,i);
break;
```

}}

TCS Ninja Coding question 5:

C Program to check whether a given number is a prime number or not. The given number N, a positive integer, will be passed to the program using the first command line parameter. If it is a prime number the output should be the square root of the number up to 2 decimal point precision, If it is not a prime number then print 0.00 to stdout.

Solution:

```
#include
#include
#include int main(int a, char *b[])
{
int number,i,flag = 1;
number = atoi(b[1]);
for(i=2; i<number; i++)
{
if(number%i == 0)
{
flag = 0;
break;
}
}
if(flag == 1)
printf("%.2f",sqrt(number));
else
printf("0.00");
return 0;
}
```

TCS Ninja Coding question 6:

is a strong number, the output should be “YES”, If it is not a prime number then output should be “NO” to stdout. Other than YES or NO, no other extra information should be printed to stdout.

Solution:

```
#include
#include
int main(int a, char *b[])
{
int number, i, temp, sum = 0, factorial = 1;
number = atoi(b[1]);
temp = number;
while(number != 0)
{
int rem = number%10;
for(i=2; i<=rem; i++)
{
factorial = factorial * i;
}
sum = sum + factorial;
number = number/10;
factorial = 1;
}
if(temp == sum)
printf(“YES”);
else
printf(“NO”);
return 0;
}
```

TCS Ninja Coding question 7:

first command line parameter. Print the equivalent binary number to stdout. Other than the binary number, no other extra information should be printed to stdout Example: Given input “19”, here N=19, expected output 10011

Solution:

```
#include
#include
int main(int a, char *argv[])
{
int number, count, i;
int b[32];
number = atoi(argv[1]);
count = 0;
while(number != 0)
{
b[count]=number%2;
number = number/2;
count++;
}
for(i=(count-1); i>=0; i–)
printf(“%d”, b[i]);
return 0;
}
```

TCS Ninja Coding question 8:

Write a c program that will find the sum of all prime numbers in a given range. The range will be specified as command line parameters. The first command line parameter, N1 which is a positive integer, will contain the lower bound of the range. The second command line

parameter N2, which is also a positive integer will contain the upper bound of the range. The

and “24” here N1= 7 and N2=24, expected output as 83.

Solution:

```
#include
int main(int argc, char *argv[])
{
int N1, N2, j, i, count, sum = 0;
N1 =atoi(argv[1]);
N2 =atoi(argv[2]);
for(i=N1+1; i<N2; ++i)
{
count = 0;
for(j=2; j<=(i/2); j++)
{
if(i%j==0)
{
count++;
break;
}
}
if(count==0)
sum = sum + i;
}
printf(“%d”,sum);
return 0;
}
```

TCS Ninja Coding question 9:

Write a C program to check whether the given number is a perfect square or not using

```
#include
#include
int main(int a, char *b[])
{
int n, i;
n= atoi(b[1]);
for(i = 0; i <= n; i++)
{
if(n == i * i)
{
printf("YES");
return 0;
}
}
printf("NO");
return 0;
}
```

TCS Ninja Coding question 10:

Write a C program to check whether the given number is Palindrome or not using command line arguments.

Solution:

```
#include
#include
int main(int a,int *b[])
{
```

```
int  number, rem, sum = 0;
```

```
{
rem =number%10;
sum = sum * 10 + rem;
number = number/10;
}
if(copy == sum)
printf(“Palindrome”);
else
printf(“Not Palindrome”);
return 0;
}
```

TCS Ninja Coding question 11:

Write a C program to convert the vowels to an uppercase in a given string using command line arguments.

Example: if the input is tata, then the expected output is tAtA.

Solution:

```
#include
int main(int argc, char *argv[])
{
char *str = argv[1];
int i;
for(i =0; str[i] !=’\0′; i++)
{
if(str[i] == ‘a’ || str[i] == ‘e’ || str[i] == ‘i’ || str[i] == ‘o’ || str[i] == ‘u’)
{
```

```
str[i] = str[i] – 32;
}



return 0;
}
```

TCS Ninja Coding question 12:

Write a C program to find the hypotenuse of a triangle using command line arguments.

Solution:

```
#include int main(int a, char*b[])
{
float hyp;
int opp=atoi(b[1]);
int adj=atoi(b[2]);
hyp=sqrt((opp*opp)+(adj*adj));
printf("%0.2f",hyp);
return 0;
}
```

TCS Ninja Coding question 13:

Write a C program to find whether the given number is an Armstrong number or not using command line arguments.

An Armstrong number of three digits is an integer such that the sum of the cubes of its digits is equal to the number itself. For example, 371 is an Armstrong number since 3**3 + 7**3 + 1**3 = 371.

Solution:

```
#include
```

```
int n;
n= atoi(b[1]);
int sum=0;
int temp=n;
int cnt=0;
while(n!=0)
{
n=n/10;
cnt++;
}
n=temp;
while(n!=0)
{
int rem=n%10;
sum=sum+pow(rem,cnt);
n=n/10;
}
if(temp==sum)
{
printf(“yes”);
}
else
{
printf(“no”);
}
return 0;
}
```

TCS Ninja Coding question 14:

Write a program to generate Fibonacci Series.

```
#include
#include
int main(int a, char *b[])
{
int i, n, t1 = 0, t2 = 1, nextTerm;
n=atoi(b[1]);
for (i = 1; i <= n; ++i)
{
printf("%d ", t1);
nextTerm = t1 + t2;
t1 = t2;
t2 = nextTerm;
}
return 0;
}
```