Coding Section TCS Questions

Coding Section TCS Questions

1. Write a C program to swapping of two numbers without using third variable. The two numbers are an integer that will be passed to the program as the first and second command line parameter. Write the output to stdout.

   Example: Given input 3, 5, expected output 5, 3.

Coding Section TCS Questions

```
#include<stdio.h>
#include<stdlib.h>
void main(int argc, char *argv[ ])
{
        int a, b;
        a = atoi(argv[1]);
        b = atoi(argv[2]);
        a = a + b;
        b = a - b;
        a = a - b;
        printf("%d %d",a, b);
}
```

2. Write a C program to calculate the factorial of a non-negative integer N. The factorial of a number N is defined as the product of all integers from 1 up to N. Factorial of 0 is defined to be 1. The number N is a non-negative integer that will be passed to the program as the first command line parameter. Write the output to stdout formatted as an integer WITHOUT any other additional text. You may assume that the input integer will be such that the output will not exceed the largest possible integer that can be stored in an int type variable.

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[ ])
            // command line arguments
{
    int N, i, fact=1;
    N = atoi(argv[1]);
            //atoi() func is to convert a string to integer
    for(i=1;i<=N;i++)
    fact=fact * i;
    printf("%d",fact);
    return 0;
}
```

Coding Section TCS Questions

3. Write a C program to find the area of a triangle given the base and the corresponding height. The values base (b) and height (h) are both positive integers passed to the program as the first and second command line parameters respectively. Write the output to stdout formatted as a floating point number rounded to EXACTLY 2 decimal precision WITHOUT any other additional text. Scientific format (such as 1.00E+5) should NOT be used while printing the output. You may assume that the inputs will be such that the output will not exceed the largest possible real number that can be stored in a float type variable. For example, given input “5” and “5”, here b = 5 and h = 5, expected output 12.50.

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[ ])
            // command line arguments
{
     int b,h;
     float area;
     b = atoi(argv[1]);
                 //atoi() func is to conv. a string to integer
     h = atoi(argv[2]);
     area = b * h/2.0;
     printf("%0.2f",area);
     return 0;
}
```

Coding Section TCS Questions

4. Write a C program to find the Hypotenuse of a right angled triangle given the two other sides of the triangle. The values of two other sides of the triangle are positive integers passed to the program as the first and second command line parameters respectively. Write the output to stdout formatted as a floating point number rounded to EXACTLY 2 decimal precision WITHOUT any other additional text. Scientific format (such as 1.00E+5) should NOT be used while printing the output. You may assume that the inputs will be such that the output will not exceed the largest possible real number that can be stored in a float type variable. For example, given input “3” and “4”, expected output 5.00.

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[ ])
            // command line arguments
{
    int a,b;
    float c;
    a = atoi(argv[1]);
            //atoi() func is to conv. a string to integer
    b = atoi(argv[2]);
    c = sqrt(a * a + b * b);
    printf("%0.2f",c);
    return 0;
}
```

Coding Section TCS Questions

5. Write a C program, to find the area of a circle when the diameter is given, using command line arguments. The input diameter is an integer. Write the output to stdout formatted as a floating point number rounded to EXACTLY 2 decimal precision WITHOUT any other additional text. Scientific format (such as 1.00E+5) should NOT be used while printing the output.

For example, given input “5”, expected output 19.62.

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[ ])
            // command line arguments
{
    int d;
    float area;
    d = atoi(argv[1]);
            //atoi() func is to conv. a string to integer
    area = 3.14 *(d/2.0)*(d/2.0);
    printf("%0.2f",area);
    return 0;
}
```

Coding Section TCS Questions

6. Write a C program which will check whether a given year YYYY is a leap year. A leap year is exactly divisible by 4 except for century years(years ending with 00). The century year is a leap year only if its perfectly divisible by 400. The given year YYYY will be passed to the program using the first command line parameter. If the given year YYYY is a leap year then print YES to stdout. If the given year is not a leap year, then print NO to stdout. Note that the words YES and NO have to be printed in UPPER CASE(capital letters) Other than the word YES or No, no other extra information should be printed to stdout.

Example: Given input “1900”, here YYYY is 1900, expected output is NO

Coding Section TCS Questions

```
#include < stdio.h>
#include<stdlib.h>
void main(int argc, char *argv[ ])
{
    unsigned int year;
    year = atoi(argv[1]);
    if ((year % 400==0) || ((year % 100 !=0)
                        && (year % 4==0 )))
    printf("YES");
    else
    printf("NO");
}
```

Coding Section TCS Questions

7. Write a C program, to check whether a given number is a prime number or not using command line arguments? If the given number is a prime number then print YES to stdout. If the given number is a not prime number, then print NO to stdout. Note that the words YES and NO have to be printed in UPPER CASE(capital letters). Other than the word YES or NO, no other extra information should be printed to stdout.

Example: Given input 13, expected output is YES

```
#include<stdio.h>
#include<math.h>
#include<stdlib.h>
void main(int argc,
                char *argv[ ])
{
     unsigned int n, i ,x;
     unsigned char flag = 0;
     n=atoi(argv[1]);
     x = sqrt(n);
     for (i = 2; i <= x ; i++)
     {
          if(n%i==0)
                {flag = 1;break;}
     }
     if(flag == 1)
     printf("NO");
     else
     printf("YES");
} /* End of main() function*/
```

8. Write a C program, to check whether a given number is a prime number or not (without using sqrt() library function) using command line arguments? If the given number is a prime number then print YES to stdout. If the given number is a not prime number, then print NO to stdout. Note that the words YES and NO have to be printed in UPPER CASE(capital letters) Other than the word YES or No, no other extra information should be printed to stdout.

Example: Given input 13, expected output is YES

```
#include<stdio.h>
#include<stdlib.h>
void main(int argc, char *argv[ ])
{
      unsigned int n, i ,x;
      unsigned char flag = 0;
      n=atoi(argv[1]);
      for (i = 2; i <= n/2 ; i++)
      {
            if(n%i==0){flag = 1;break;}
      }
      if(flag == 1)
      printf("NO");
      else
      printf("YES");
} /* End of main() function*/
```

Coding Section TCS Questions

9. Write a C program, to check whether a given number is a prime number or not using command line arguments? If the given number is a prime number, print it's square root as floating point number rounded to EXACTLY 2 decimal precision WITHOUT any other additional text to stdout. If the given number is a not prime number, print 0.00 to stdout.

Example: Given input 13, expected output is 3.61

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
void main(int argc,
                char *argv[ ])
{
     unsigned int n, i ;
     float x;
     unsigned char flag = 0;
     n=atoi(argv[1]);
     x = sqrt(n);
     for (i = 2; i <= (int)x ; i++)
     {
          if(n%i==0)
               {flag = 1;break;}
     }
     if(flag == 1)
     printf("%0.2f",x);
     else
     printf("%0.2f",0);
} /* End of main() function*/
```

Coding Section TCS Questions

10. Write a C program, to check whether a given number is a perfect square or not using command line arguments? If the given number is a perfect square then print YES to stdout. If the given number is a not perfect square, then print NO to stdout. Note that the words YES and NO have to be printed in UPPER CASE(capital letters) Other than the word YES or NO, no other extra information should be printed to stdout.

Example: Given input 16, expected output is YES

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
void main(int argc, char *argv[ ])
{
      unsigned int n,a;
      n = atoi(argv[1]);
      a = sqrt (n);
        /*Type conversion (float value convert into integer value)*/
      if( a*a == n )
      printf("YES");
      else
      printf("NO");
}
```

Coding Section TCS Questions

11. Write a C program, to check whether a given number is a perfect square or not (without using sqrt() library function) using command line arguments? If the given number is a perfect square then print YES to stdout. If the given number is a not perfect square, then print NO to stdout. Note that the words YES and NO have to be printed in UPPER CASE(capital letters) Other than the word YES or NO, no other extra information should be printed to stdout.

Example: Given input 16, expected output is YES

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
void main(int argc,
                char *argv[ ])
{
      unsigned int n,i;
      unsigned char flag = 0;
      unsigned long x;
      n = atoi(argv[1]);
      for(i=0; (x=i * i)<=n; i++)
      {
            if( x == n)
            {
                  flag = 1; break;
            } // End of if
        } // End of for loop
      if(flag == 1)
      printf("YES");
      else
      printf("NO");
} /* End of main function */
```

Coding Section TCS Questions

12. Write a C program to find the sum of square of N given numbers. The value N and given numbers will be specified as command line parameters. Write the output to stdout formatted as an integer WITHOUT any other additional text.

Example: Given input 5, 1, 3, 4 ,7, 5 expected output 100.

```
#include<stdio.h>
#include<stdlib.h>
void main(int argc, char *argv[ ])
{
    unsigned int n, i;
    int temp, sum =0;
    n = atoi(argv[1]);
    for (i=1;i<=n; i++)
    {
        temp=atoi(argv[ i+1]);
        sum = sum + temp * temp;
    }
    printf (“%d”, sum);
}
```

Coding Section TCS Questions

13. Write a C program to find the sum of digits of a non-negative integer. The number is a non-negative integer that will be passed to the program as the first command line parameter. Write the output to stdout formatted as an integer WITHOUT any other additional text.

Example: Given input 123, expected output is 6.

```
#include<stdio.h>
#include<stdlib.h>
void main(int argc, char *argv[ ])
{
    unsigned int sum=0, num;
    num = atoi(argv[1]);
    while( num > 0)
    {
        sum = sum + num % 10;
        num = num / 10;
    }
    printf ("%u", sum);
}
```

Coding Section TCS Questions

14. Write a C program to find the reverse of a non-negative integer. The number is a non-negative integer that will be passed to the program as the first command line parameter. Write the output to stdout formatted as an integer WITHOUT any other additional text.

Example: Given input 123, expected output is 321.

```
#include<stdio.h>
#include<stdlib.h>
void main(int argc, char *argv[ ])
{
    unsigned int num, rev=0;
    num = atoi(argv[1]);
    while( num >0)
    {
        rev = rev * 10 + num % 10;
        num = num / 10;
    }
    printf ("%u", rev);
}
```

15. Write a C program, to check whether a given 3 digit number is a palindrome number or not using command line arguments? If the given number is a palindrome number then print YES to stdout. If the given number is a not palindrome number, then print NO to stdout. Note that the words YES and NO have to be printed in UPPER CASE(capital letters) Other than the word YES or No, no other extra information should be printed to stdout.

Example: Given input 121, expected output is YES

Coding Section TCS Questions

```
#include<stdio.h>
#include<stdlib.h>
void main(int argc, char *argv[ ])
{
    unsigned int num;
    num = atoi(argv[1]);
    if ( num % 10 == num/100)
    printf ("YES");
    else
    printf ("NO");
}
```

16. Write a C program, to check whether a given number is a palindrome number or not using command line arguments? If the given number is a palindrome number then print YES to stdout. If the given number is a not palindrome number, then print NO to stdout. Note that the words YES and NO have to be printed in UPPER CASE(capital letters) Other than the word YES or No, no other extra information should be printed to stdout.

Example: Given input 1221, expected output is YES

```
#include<stdio.h>
void main(int argc, char *argv[ ])
{
    unsigned int num, rev=0, numtemp ;
    num = atoi(argv[1]);
    numtemp = num;
    while( num >0)
    {
        rev = rev * 10 + num % 10;
        num = num / 10;
    }
    if ( numtemp == rev)
    printf ("YES");
    else
    printf ("NO");
}
```

17. Write a C program, to check whether a given number is an Armstrong number or not using command line arguments? If the given number is an Armstrong number then print YES to stdout. If the given number is not an Armstrong number, then print NO to stdout. Note that the words YES and NO have to be printed in UPPER CASE (capital letters) Other than the word YES or NO, no other extra information should be printed to stdout.

Example: Given input 371, expected output is YES.

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
void main(int argc,
                char *argv[ ])
{
     unsigned int num,
              sum=0, numtemp;
     unsigned char count=0;
     num = atoi(argv[1]);
     numtemp = num;
     while( num >0)
     { / * Start of while loop*/
          count += 1;
          num = num / 10;
     } /* End of while loop*/
     num = numtemp;
     while( num >0)
     { / * Start of while loop*/
          sum = sum +  pow
            ((num % 10), count);
          num = num / 10;
     } * End of while loop*/
     if ( numtemp == sum)
     printf ("YES");
     else
     printf ("NO");
}
```

Coding Section TCS Questions

18. Write a C program, to find the factor of the given non-negative number using command line argument. The input is non-negative integers. Write the output to stdout formatted as an integer WITHOUT any other additional text.

Example: Given input 9 expected output is 3 3.

```
#include<stdio.h>
#include<stdlib.h>
void main(int argc,
                char *argv[ ])
{
     unsigned int n, i;
     n = atoi(argv[1]);
     i = 2;
     while ( i * i <= n)
     {
          if ( n % i == 0)
          {
               printf ("%u ", i);
               n = n / i;
          }
          else
          i  = i +1;
     }
     printf ("%u", n);
}
```

Coding Section TCS Questions

19. Write a C program, to find the GCD (HCF) of the given 2 numbers, using command line arguments. The input is 2 integers. Write the output GCD (HCF) to stdout formatted as an integer WITHOUT any other additional text.

Example: Given input 8, 12, expected output is 4.

```
void main(int argc,
                char *argv[ ])
{
    unsigned int a, b, rem;
    a = atoi(argv[1]);
    b= atoi(argv[2]);
    do
    {
        rem = a % b;
        if ( rem == 0) break;
        a = b;
        b = rem;
    }
    while  (1);
    printf ("%u", b);
}
```

Coding Section TCS Questions

20. Write a C program, to find the LCM of the given 2 numbers, using command line arguments. The input is 2 integers. Write the output LCM to stdout formatted as an integer WITHOUT any other additional text.

Example: Given input 8, 12, expected output is 24.

```
#include<stdio.h>
void main(int argc,
                char *argv[ ])
{
    unsigned int a, b,c, rem;
    a = atoi(argv[1]);
    b= atoi(argv[2]);
    c= a*b;
    do
    {
        rem = a % b;
        if ( rem == 0)break;
        a = b;
        b = rem;
    }
    while  (1);
    printf ("%u ", c/b);
}
```

Coding Section TCS Questions

21. Write a C program which will convert a given decimal integer number N to its binary equivalent. The given number N, a positive integer, will be passed to the program using the 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.

```
#include<stdio.h>
#include<stdlib.h>
void main(int argc,
                char *argv[ ])
{
     unsigned int i=0,num;
     unsigned char
                    a[sizeof(int)];
     int j;
     num= atoi(argv[1]);
     while( num != 0)
     {
          a[i++]= num%2;
          num = num / 2;
     }
     for(j=i-1;j >=0; j--)
     printf("%u",a[j]);
}
```

Coding Section TCS Questions

22. Write a C program to find the Fibonacci series up to N number of terms. The value of N will be specified as command line parameters, which is a positive integer. Write the output to stdout WITHOUT any other additional text.

Example: Given input “5”, expected output is 1 1 2 3 5

```
#include<stdio.h>
#include<stdlib.h>
void main(int argc,
                char *argv[ ])
{
     unsigned int n,a=1,
                        b=1,i=1,c;
     n=atoi(argv[1]);
     printf ("%u ", a);
     while( i++ < n)
     {
          printf ("%u ",b);
          c = a + b;
          a = b;
          b = c;
     }
}
```

23. Write a C program to print Floyd's triangle up to N number of rows. The value of N will be specified as command line parameters, which is a positive integer. Write the output to stdout.

Example: Given input 3, expected output.

1

23

456

```
#include<stdio.h>
#include<stdlib.h>
void main(int argc, char *argv[ ])
{
    unsigned char row, col, n,i,count=1;
    n=atoi(argv[1]);
    for(row=1;row <= n; row++)
    {
        for(col=1;col <= row; col++)
        printf("%d ",count++);
        printf("\n");
    }
}
```

24. Write a C program to sum of two numbers and print the sum in binary (convert a sum of two decimal integer numbers to its binary equivalent). The given two numbers, a positive integer, will be passed to the program using the first and second 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 are 7 and 12, expected output 10011.

```
#include<stdio.h>
#include<stdlib.h>
void main(int argc,
                char *argv[ ])
{
    unsigned int i=0,num;
    unsigned char
                a[sizeof(int)];
    int j;
    num= atoi(argv[1])
            + atoi(argv[2]);
    while( num != 0)
    {
        a[i++]= num%2;
        num = num / 2;
    }
    for(j=i-1;j >=0; j--)
    printf("%u",a[j]);
}
```

Coding Section TCS Questions

25. Write a C program that will find the sum of all odd 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 program should consider all the odd numbers within the range, excluding the upper bound and lower bound. Print the output in integer format to stdout. Other than the integer number, no other extra information should be printed to stdout.

Example: Given inputs “2” and “7” here N1=2 and N2=7, expected output as 8

```
#include<stdio.h>
#include<stdlib.h>
void main(int argc, char *argv[ ])
{
    unsigned int n1,n2, i ,sum=0;
    n1=atoi(argv[1]);
    n2=atoi(argv[2]);
    for (i = n1+1; i < n2 ; i++)
    if(i%2 != 0)
    sum= sum + i;
    printf("%d",sum);
}
```

Coding Section TCS Questions

26. Write a C program that will find the average of all odd 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 program should consider all the odd numbers within the range, excluding the upper bound and lower bound. Write the output to stdout formatted as a floating point number rounded to EXACTLY 2 decimal precision WITHOUT any other additional text. Scientific format (such as 1.00E+5) should NOT be used while printing the output.

Example: Given inputs “2” and “7” here N1=2 and N2=7, expected output as 4.00.

```
#include<stdio.h>
#include<stdlib.h>
void main(int argc, char *argv[ ])
{
    unsigned int n1,n2, i ,sum=0,count=0;
    n1=atoi(argv[1]);
    n2=atoi(argv[2]);
    for (i = n1+1; i < n2; i++)
    if(i%2 != 0)
    {
        sum= sum + i;
        count++;
    }
    printf("%.2f",(float)sum/count);
}
```

Coding Section TCS Questions

27. Write a C program that will find the average of all even numbers in between 1 to N. The value of N will be specified as command line parameters. The first command line parameter, N which is a positive integer, will contain the upper bound of the range. The program should consider all the even numbers within the range, including the upper bound and lower bound. Write the output to stdout formatted as a floating point number rounded to EXACTLY 2 decimal precision WITHOUT any other additional text. Scientific format (such as 1.00E+5) should NOT be used while printing the output.

Example: Given inputs 8, here N=8, expected output as 5.00.

```
#include<stdio.h>
#include<stdlib.h>
void main(int argc, char *argv[ ])
{
    unsigned int n, i, sum=0, count=0;
    n=atoi(argv[1]);
    for (i = 1; i <= n; i++)
    if(i%2 == 0)
    {
        sum=sum + i;
        count++;
    }
    printf("%.2f",(float)sum/count);
}
```

28. Write a C program which will convert a given binary number to its decimal integer equivalent. The given binary number, will be passed to the program using the first command line parameter. Print the equivalent decimal number to stdout. Other than the decimal number, no other extra information should be printed to stdout.

Example: Given input “10011”, expected output 19.

Coding Section TCS Questions

```
#include<stdio.h>
#include<stdlib.h>
void main(int argc, char *argv[ ])
{
    unsigned int num, decimaln=0,i=1;
    num=atoi(argv[1]);
    while( num != 0)
    {
        decimaln += num %10 * i;
        i *= 2;
        num = num / 10;
    }
    printf ("%u", decimaln);
}
```

Coding Section TCS Questions

29. 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 program should consider all the prime numbers within the range, excluding the upper bound and lower bound. Print the output in integer format to stdout. Other than the integer number, no other extra information should be printed to stdout.

Example: Given inputs “7” and “24” here N1=7 and N2=24, expected output as 83.

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc,
            char *argv[])
{
    int i,j,flag,sum=0,n1,n2;
    n1=atoi(argv[1]);
    n2=atoi(argv[2]);
    for(i=n1+1;i<n2;i++)
    {
        flag=0;
        for(j=2;j<=i/2;j++)
        {
        if((i%j)==0)
        {
            flag=1; break;
        }
    }
    if(flag==0)
    sum += i;}
    printf("%d",sum);
    return 0;
}
```

Coding Section TCS Questions

30. Write a C program, to find the length of string. The string will be passed to the program as the first command line parameter. Note that the length of string has to be printed to stdout. Other than the length of string, no other extra information should be printed to stdout.

Example: Given input “abc”, expected output is 3.

Coding Section TCS Questions

```
#include<stdio.h>
#include<stdlib.h>
void main(int argc, char *argv[ ])
{
    int j=0, i =0;
    while(argv[1][i++] != '\0');
    printf ("%d", i-1);
}
```

Coding Section TCS Questions

31. Write a C program, to find the reversal of string. The string will be passed to the program as the first command line parameter. Note that the reverse of string has to be printed to stdout. Other than the reverse of string, no other extra information should be printed to stdout.

Example: Given input “abc”, expected output is “cba”.

```
#include<stdio.h>
#include<stdlib.h>
void main(int argc,
                char *argv[ ])
{
    int j=0, i =0;
    char *b;
    while(argv[1][i++] != '\0');
    i=i-2;
    b= (char*) malloc((i+1)
                * sizeof(char));
    while(i>=0)
    b[j++] = argv[1][i--];
    b[j]='\0';
    printf ("\n %s \n", b);
    free(b);
}
```

32. Write a C program, to check whether a given string is a palindrome or not using command line arguments? If the given string is a palindrome then print YES to stdout. If the given string is a not palindrome, then print NO to stdout. Note that the words YES and NO have to be printed in UPPER CASE (capital letters) Other than the word YES or No, no other extra information should be printed to stdout.

Example: Given input “aba”, expected output is YES

```
#include<stdio.h>
#include<stdlib.h>
void main(int argc,
                char *argv[ ])
{
    int j=0, i =0;
    char b[10],flag=0;
    while(argv[1][i++] != '\0');
    i=i-2;
    while(i>=0)
    b[j++] = argv[1][i--];
    b[j]='\0';
    i=0;
    while(argv[1][i] != '\0'
                    || b[i] != '\0')
    if(argv[1][i]!= b[i++])
            {flag=1;break;}
    if(flag==0)
    printf(“YES");
    else
    printf("NO");
}
```

Coding Section TCS Questions

33. Write a C program, to convert string from upper case to lower case. The string will be passed to the program as the first command line parameter. Note that the lower case string has to be printed to stdout. Other than the string, no other extra information should be printed to stdout.

Example: Given input “ABC”, expected output is “abc”.

Coding Section TCS Questions

```
#include<stdio.h>
#include<stdlib.h>
void main(int argc, char *argv[ ])
{
    int i=0;
    while(argv[1][i] != '\0')
    {
        if(argv[1][i] >='A' && argv[1][i] <= 'Z')
        argv[1][i]= argv[1][i] + 32;
        i++;
    }
    printf("%s",argv[1]);
}
```

34. Write a C program, to convert string from lower case to upper case. The string will be passed to the program as the first command line parameter. Note that the upper r case string has to be printed to stdout. Other than the string, no other extra information should be printed to stdout.

Example: Given input “abc”, expected output is “ABC”.

Coding Section TCS Questions

```
#include<stdio.h>
#include<stdlib.h>
void main(int argc, char *argv[ ])
{
    int i=0;
    while(argv[1][i] != '\0')
    {
        if(argv[1][i] >='a' && argv[1][i] <= 'z')
        argv[1][i]= argv[1][i] - 32;
        i++;
    }
    printf("%s",argv[1]);
}
```

35. Write a C program, to check whether a given number is an even number or not using command line arguments? If the given number is an even number then print YES to stdout. If the given number is not an even number, then print NO to stdout. Note that the words YES and NO have to be printed in UPPER CASE (capital letters) Other than the word YES or NO, no other extra information should be printed to stdout.

Example: Given input 7, expected output is NO.

Coding Section TCS Questions

```
#include<stdio.h>
#include<stdlib.h>
void main(int argc, char *argv[ ])
{
    unsigned int num ;
    num = atoi(argv[1]);
    if(num%2 == 0)
    printf("YES");
    else
    printf("NO");
}
```

36. Write a C program to print multiplication table of a non-negative integer N. The number N is a non-negative integer that will be passed to the program as the first command line parameter. Write the output to stdout formatted.

Example: Given input “3”, expected output 3 6 9 12 15 18 21 24 27 30.

Coding Section TCS Questions

```
#include<stdio.h>
#include<stdlib.h>
void main(int argc, char *argv[ ])
{
      unsigned int n,i ;
      n = atoi(argv[1]);
      for(i=1;i<=10;i++)
      printf("%d ", i*n);
}
```

Coding Section TCS Questions

37. Write a C program to convert temperature from degree centigrade to Fahrenheit. The value is floating point number passed to the program as the first command line parameter. Write the output to stdout formatted as a floating point number rounded to EXACTLY 2 decimal precision WITHOUT any other additional text. Scientific format (such as 1.00E+5) should NOT be used while printing the output. You may assume that the inputs will be such that the output will not exceed the largest possible real number that can be stored in a float type variable.

Example: Given input “31”, expected output 87.80.

Coding Section TCS Questions

(Formula: T(°F) = T(°C) × 9/5 + 32)

```
#include<stdio.h>
#include<stdlib.h>
void main(int argc, char *argv[ ])
{
    float  tc;
    tc = atof(argv[1]);
    printf("%.2f ",tc*(9/5.0)+32 );
}
```

Coding Section TCS Questions

38. Write a C program to convert temperature from Fahrenheit to degree centigrade. The value is floating point number passed to the program as the first command line parameter. Write the output to stdout formatted as a floating point number rounded to EXACTLY 2 decimal precision WITHOUT any other additional text. Scientific format (such as 1.00E+5) should NOT be used while printing the output. You may assume that the inputs will be such that the output will not exceed the largest possible real number that can be stored in a float type variable.

Example: Given input “103”, expected output 39.44

Coding Section TCS Questions

(Formula: T(°C) = (T(°F) - 32) / (9/5))

```
#include<stdio.h>
#include<stdlib.h>
void main(int argc, char *argv[ ])
{
    float  tf ;
    tf = atof(argv[1]);
    printf("%.2f ",(tf-32)/(9/5.0) );
}
```

Coding Section TCS Questions

39. Write a C program to find the largest among ten numbers. The value of numbers will be specified as command line parameters. Write the output to stdout.

Example: Given input 3, 6, 5, 8, 12, 11, 14, 7, 15, 13, expected output 15.

Coding Section TCS Questions

```
#include<stdio.h>
#include<stdlib.h>
void main(int argc, char *argv[ ])
{
    unsigned int i;
    int largest = 0;
    for ( i = 1;i <= 10 ; i++)
    if (largest < atoi(argv[i]))
    largest =atoi(argv[i]);
    printf("%d",largest);
}
```

40. Write a C program to find the second largest among ten numbers. The value of numbers will be specified as command line parameters. Write the output to stdout.

Example: Given input 3, 6, 5, 8, 12, 11, 14, 7, 15, 13, expected output 14.

```
#include<stdio.h>
#include<stdlib.h>
void main(int argc, char *argv[ ])
{
        int array[10] ;
        int i, j, temp;
        for(i = 1; i <= 10; i++)
        array[i-1]=atoi(argv[i]);
        for(i = 1; i < 10; i++)
        {
            for (j=0; j < 10-i;j++)
            if (array[j] > array[j+1])
                { /* Interchange the
                     data items */
                     temp = array[j];
                     array[j] = array[j+1];
                     array[j+1] = temp;
                } /* End of if */
        } /* End of outer for loop */
        printf("%d", array[8]);
}
```

Q41. Write a C program to print Pascal's triangle up to N number of rows. The value of N will be specified as command line parameters, which is a positive integer. Write the output to stdout. Example: Given input 3, expected output.

```
  1
 121
12321
```

Q42. Write a C program, to check whether a given number is a perfect number or not using command line argument. Perfect number, a positive integer that is equal to the sum of its proper divisors. The smallest perfect number is 6, which is the sum of 1, 2, and 3. Example: Given input 28, expected output is YES

Q43. Write a C program, to check whether a given number is a strong number or not using command line argument. A positive integer number is called strong number if sum of the factorial of its digit is equal to number itself. Example: Given input 145, expected output is YES.

(For example: 145 since 1! + 4! + 5! = 1 + 24 + 120 = 145)

Q44. Write a C program to find roots of a quadratic equation. The value will be specified as command line parameters.

Q45. Write a C program to count vowels and consonants in the given string.

Q46. Consider the following series:

1,1,2,3,4,9,8,27,16,81,32,243,64,729,128,2187…

This series is a **mixture of 2 series - all the odd terms in this series form a geometric series** and **all the even terms form yet another geometric series. Write a program to find the Nth term in the series.**

The value N in 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 value of nth term, no other character / string or message should be written to STDOUT.

For example , if N=16, the 16th term in the series is 2187, so only value 2187 should be printed to STDOUT.

You can assume that N will not exceed 30.

Example-4

Consider the following 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** prime numbers in ascending order. **Write a program to find the Nth term in the series.** The value N in 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 value of nth term, no other character / string or message should be written to STDOUT.

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

You can assume that N will not exceed 50.

Q47. Consider the below series :

0,0,2,1,4,2,6,3,8,4,10,5,12,6,14,7,16,8

This series is a mixture of 2 series all the odd terms in this series form even numbers in ascending order and every even terms is derived from the previous term using the formula (x/2)

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

The value n in 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 the nth term no other characters /strings or message should be written to STDOUT.

Q48. Write a program such that to get three strings from user in each new line, replace the vowels in the first string with $, replace the consonants in the second string with # and convert the third string to uppercase.

Input Format

Three strings, each in a new line.

Output Format

Display the output as mentioned in the question.

Constraints

strings(s1,s2,s3) != 0;

Sample Input

love
for
life

Sample Output

lv#o#LIVE

Q49. Given an array **A** consist of **N** number of elements. If the sum of the element is "**even**" print the **sum** of the element. If the sum of the element is **"odd"** print the **product** of the element.

Input Format

The first line of input contains the number of elements **N.**

The second line of input represents the elements $\mathbf{A_1}$, $\mathbf{A_2}$, $\mathbf{A_3}$ . . . $\mathbf{A_N}$

Output Format

Prints the desired result

Sample 1 Input

5 1 2 3 4 5

Sample 1 Output

120

Sample 2 Input

4 10 20 52 51

Sample 2 Output

530400

Q50. For example if n=10,the 10^{th} term in the series is to be derived from the 9th term in the series. The 9th term is 8 so the 10th term is (8/2)=4. Only the value 4 should be printed to STDOUT.

You can assume that the n will not exceed 20,000.