```

########################### EX 1 ###################################

/* addition program
 * it reads two integer numbers and prints their sum
 * */

#include<stdio.h>

int main(){
   int number1, number2, sum;   /* declaration of variables */

   printf("Enter first number\n");  /*  prompt message on console */
   scanf("%d", &number1); /* read an integer */
   printf("Enter another number\n"); /* prompt again a message */
   scanf("%d", &number2); /* read another integer */
   sum=number1+number2;   /* assignment of sum */
   printf("Sum is %d\n", sum); /* the sum value is printed */

   return 0; /* return from main */

}

########################### EX 2 ###################################

/* demonstrates the input and output for a char
 * note that the character is stored as an integer (int, not char)
 * and is read and printed with %c
 * */

#include<stdio.h>
   int main(){
   int c;  // declare the variable that will store the value inserted
   scanf(" %c",&c); // read the character
   printf("%c\n",c); // print the character

/* alternative way of reading writing characters
 * c=getchar();
   putchar(c);
*/
   return 0; // return from main

}


########################### EX 3 ###################################
# include <stdio.h>

int main()
{
    int i, num, j;
    printf ("Enter the number: ");
    scanf ("%d", &num );

    for (i=1; i<num; i++)
        j=j*i;

    printf("The factorial of %d is %d\n",num,j);
}


########################### EX 4 ###################################

/* example of using float values */
#include<stdio.h>

int main(){
    float number1, number2;
        double dnumber1, dnumber2;

    number1=25.223f;
    printf("Input a float value:");
    scanf(" %f", &number2);
    printf("\nThe values are %6.4f, %6.4f\n",number1, number2);

        dnumber1=25.223;
        printf("Input a double value:");
        scanf(" %lf", &dnumber2);
        printf("\nThe values are %6.4lf, %6.4lf\n",dnumber1, dnumber2);


    return 0;
}

########################### EX 5 ####################################

/* computes the greatest common divisor */
#include<stdio.h>
int main(){
   int m, n, r;
   do{
     printf("\nEnter two positive integers:");
     scanf("%d %d",&m, &n);
   }while(m<=0||n<=0);
   do{
     r=m%n;
     m=n;
     n=r;
   }while(r>0);
   printf("result is %d\n",m);
   return 0;
}

########################### EX 6 ####################################

/* print a multiplication table */
#include<stdio.h>
int main(){
   int type, start, end, j;
   printf("Type of table?");
   scanf("%d",&type);
   printf("start of table?");
   scanf("%d",&start);
   printf("end of table?");
   scanf("%d",&end);
   for(j=start;j<=end;j++)
      printf("\n%2d x %2d = %3d", j, type, j * type);
   printf("\nEnd of program\n");
   return 0;
}

########################### EX 7 ####################################

/* gets() example */
#include <stdio.h>

int main()
{
    char sir[20];

    printf("Introduceti un string: ");
    gets(sir);
    printf("Sirul este: %s\n", sir);

    return 0;
}


```

```
########################### EX 8 ###################################
/* detects if a number is a perfect square or not
 * uses the method of checcking if its square root is a integer value or not
 */

#include <stdio.h>
#include <math.h>  // library needed for sqrt function; compilation should be done with "-lm" option

int main(){
    int number;
    printf ("\n Introduce an integer: ");  // display a message
    scanf ("%d", &number);                 // read a integer value

    if (sqrt(number)==floor(sqrt(number))) // check if its square root is integer
        printf ("\n The integer is a perfect square \n");
    else
        printf ("\n The integer is not a perfect square \n");

    return 0;

}

########################### EX 9 ###################################

/* detect if the given value reresents a prime number or not
 * it uses the method for checking all values less than the given number for being a divisor
 */
#include <stdio.h>

int main(){

    int primed, number, flag;

    flag = 1;         // suppose it is prime
    printf("Give a positive integer:");
    scanf("%d", &number);

    for (primed=2; primed < number; primed++)
    {
        if ((number%primed)==0)
        flag = 0; // current primed is a divisor because no remainder, so is not prime
    }
    if(flag == 1)
        printf("\n number %d is prime\n", number);  //all values tested and all have remainder so the number is prime
    else
        printf("\n number %d is not prime\n", number);
    return 0;
}


########################### EX 10 ###################################

/* demonstrates the use of special characters in output
 * note that 'b' changes its meaning by being preceeded with '\'
 */

#include <stdio.h>

int main()
{

    printf("space \bnon-space\n"); // the output is in accordance with \b (backspace)
    return(0);

}
```

```
########################### EX 11 ###################################
/* demonstrates the use of special characters in output
 * note that 'b' changes its meaning by being preceeded with '\' (backspace)
 * the same for '\t' (horisontal tab) and '\n' (new line)
 * it also demonstrates the format specifier for real numbers (float data type)
 *   */


#include <stdio.h>

int
main (void)
{
 printf("aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa\n" // the output string is writen in the source code on 4 lines
        "bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb\t"
        "cccccccccccccccccccccccccccccc\b"
        "\n");

  printf ("Two plus two is %f\n", 4.0f);
  return 0;
}


########################### EX 12 ###################################

/* Program demonstrates the use of width and precission for
 * output for different data types
 */

#include <stdio.h>
int main()
{
    int int_number=1234567890;
    float flt_number=1234567890.12345;
    char str[] = "string example";
// integer examples
    printf("%20d", int_number);
    printf("%-20d", int_number);
    printf("%20.5d", int_number);
// float examples
    printf("%20f", flt_number);
    printf("%-20f", flt_number);
    printf("%20.5f", flt_number);
// string examples
    printf("%20s\n", str);
    printf("%-20s\n", str);
    printf("%20.5s\n", str);
    printf("%-20.5s\n", str);
    return 0;
}


########################### EX 13 ###################################


                /************************/
                /*   Table of       */
                /*   Sine Function  */
                /************************/

                /* Michel Vallieres     */
                /* Written: Winter 1995 */
#include <stdio.h>
#include <math.h>

int main()
{
    int    angle_degree;
    double angle_radian, pi, value;

                    /* Print a header */
    printf ("\nCompute a table of the sine function\n\n");

                    /* obtain pi once for all */
                    /* or just use pi = M_PI, where
 *                      M_PI is defined in math.h    */
    pi = 4.0*atan(1.0);
    printf ( " Value of PI = %f \n\n", pi );

    printf ( " angle     Sine \n" );

    angle_degree=0;         /* initial angle value       */
                    /* scan over angle       */

    while (  angle_degree <= 360 )  /* loop until angle_degree > 360 */
    {
       angle_radian = pi * angle_degree/180.0 ;
       value = sin(angle_radian);
       printf ( " %3d      %f \n ", angle_degree, value );

       angle_degree = angle_degree + 10; /* increment the loop index     */
    }
    return 0;
}


########################### EX 14 ###################################

/* the program converts the given weight from kilograms to pounds */

#include<stdio.h>

int main(){

   float kg_val, pounds_val;
   printf("Enter a weight in kg: \n"); // display a message for user
   scanf("%f",&kg_val);                // read the input value as real numbersimple precission
   pounds_val=kg_val*2.2;              // compute the value in pounds
   printf("\n\n%6.2f kilos = %6.2f pounds \n", kg_val, pounds_val);  // display the input and output values
   return 0;                           // return from main

}
```