

```

1
2 ##### EX 1 #####
3
4 /* addition program
5  * it reads two integer numbers and prints their sum
6  * */
7
8 #include<stdio.h>
9
10 int main(){
11     int number1, number2, sum;    /* declaration of variables */
12
13     printf("Enter first number\n"); /* prompt message on console */
14     scanf("%d", &number1); /* read an integer */
15     printf("Enter another number\n"); /* prompt again a message */
16     scanf("%d", &number2); /* read another integer */
17     sum=number1+number2;    /* assignment of sum */
18     printf("Sum is %d\n", sum); /* the sum value is printed */
19
20     return 0; /* return from main */
21
22 }
23
24 ##### EX 2 #####
25
26 /* demonstrates the input and output for a char
27  * note that the character is stored as an integer (int, not char)
28  * and is read and printed with %c
29  * */
30
31 #include<stdio.h>
32 int main(){
33     int c; // declare the variable that will store the value inserted
34     scanf(" %c",&c); // read the character
35     printf("%c\n",c); // print the character
36
37     /* alternative way of reading writing characters
38     * c=getchar();
39     putchar(c);
40     */
41     return 0; // return from main
42
43 }
44
45 ##### EX 3 #####
46 # include <stdio.h>
47
48
49 int main()
50 {
51     int i, num, j;
52     printf ("Enter the number: ");
53     scanf ("%d", &num );
54
55     for (i=1; i<num; i++)
56         j=j*i;
57
58     printf("The factorial of %d is %d\n",num,j);
59 }
60
61 ##### EX 4 #####
62
63 /* example of using float values */
64 #include<stdio.h>
65
66
67 int main(){
68     float number1, number2;
69     double dnumber1, dnumber2;

```

```

70
71     number1=25.223f;
72     printf("Input a float value:");
73     scanf(" %f", &number2);
74     printf("\nThe values are %6.4f, %6.4f\n",number1, number2);
75
76     dnumber1=25.223;
77     printf("Input a double value:");
78     scanf(" %lf", &dnumber2);
79     printf("\nThe values are %6.4lf, %6.4lf\n",dnumber1, dnumber2);
80
81
82     return 0;
83 }
84
85 ##### EX 5 #####
86
87 /* computes the greatest common divisor */
88 #include<stdio.h>
89 int main(){
90     int m, n, r;
91     do{
92         printf("\nEnter two positive integers:");
93         scanf("%d %d",&m, &n);
94     }while(m<=0||n<=0);
95     do{
96         r=m%n;
97         m=n;
98         n=r;
99     }while(r>0);
100     printf("result is %d\n",m);
101     return 0;
102 }
103
104 ##### EX 6 #####
105
106 /* print a multiplication table */
107 #include<stdio.h>
108 int main(){
109     int type, start, end, j;
110     printf("Type of table?");
111     scanf("%d",&type);
112     printf("start of table?");
113     scanf("%d",&start);
114     printf("end of table?");
115     scanf("%d",&end);
116     for(j=start;j<=end;j++)
117         printf("\n%2d x %2d = %3d", j, type, j * type);
118     printf("\nEnd of program\n");
119     return 0;
120 }
121
122 ##### EX 7 #####
123
124 /* gets() example */
125 #include <stdio.h>
126
127 int main()
128 {
129     char sir[20];
130
131     printf("Introduceti un string: ");
132     gets(sir);
133     printf("Sirul este: %s\n", sir);
134
135     return 0;
136 }
137
138

```

```

139 ##### EX 8 #####
140 /* detects if a number is a perfect square or not
141    * uses the method of checking if its square root is a integer value or not
142    */
143
144 #include <stdio.h>
145 #include <math.h> // library needed for sqrt function; compilation should be done with
146                    "-lm" option
147
148 int main(){
149     int number;
150     printf ("\n Introduce an integer: "); // display a message
151     scanf ("%d", &number); // read a integer value
152
153     if (sqrt(number)==floor(sqrt(number))) // check if its square root is integer
154         printf ("\n The integer is a perfect square \n");
155     else
156         printf ("\n The integer is not a perfect square \n");
157
158     return 0;
159 }
160
161 ##### EX 9 #####
162
163 /* detect if the given value reresents a prime number or not
164    * it uses the method for checking all values less than the given number for being a
165    divisor
166    */
167 #include <stdio.h>
168
169 int main(){
170     int primed, number, flag;
171
172     flag = 1; // suppose it is prime
173     printf("Give a positive integer:");
174     scanf("%d", &number);
175
176     for (primed=2; primed < number; primed++)
177     {
178         if ((number%primed)==0)
179             flag = 0; // current primed is a divisor because no remainder, so is not prime
180     }
181     if(flag == 1)
182         printf("\n number %d is prime\n", number); //all values tested and all have
183         remainder so the number is prime
184     else
185         printf("\n number %d is not prime\n", number);
186     return 0;
187 }
188
189 ##### EX 10 #####
190
191 /* demonstrates the use of special characters in output
192    * note that 'b' changes its meaning by being preceeded with '\\'
193    */
194
195 #include <stdio.h>
196
197 int main()
198 {
199
200     printf("space \bnon-space\n"); // the output is in accordance with \b (backspace)
201     return(0);
202
203 }
204

```

```

205
206 ##### EX 11 #####
207 /* demonstrates the use of special characters in output
208  * note that 'b' changes its meaning by being preceeded with '\\' (backspace)
209  * the same for '\t' (horisontal tab) and '\n' (new line)
210  * it also demonstrates the format specifier for real numbers (float data type)
211  * */
212
213
214 #include <stdio.h>
215
216 int
217 main (void)
218 {
219     printf("aaaaaaaaaaaaaaaaaaaaaaaaaaaa\n" // the output string is written in the source
220           code on 4 lines
221           "bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb\t"
222           "cccccccccccccccccccccccccccccc\b"
223           "\n");
224
225     printf ("Two plus two is %f\n", 4.0f);
226     return 0;
227 }
228
229 ##### EX 12 #####
230
231 /* Program demonstrates the use of width and precission for
232  * output for different data types
233  */
234
235 #include <stdio.h>
236 int main()
237 {
238     int int_number=1234567890;
239     float flt_number=1234567890.12345;
240     char str[] = "string example";
241     // integer examples
242     printf("%20d", int_number);
243     printf("%-20d", int_number);
244     printf("%20.5d", int_number);
245     // float examples
246     printf("%20f", flt_number);
247     printf("%-20f", flt_number);
248     printf("%20.5f", flt_number);
249     // string examples
250     printf("%20s\n", str);
251     printf("%-20s\n", str);
252     printf("%20.5s\n", str);
253     printf("%-20.5s\n", str);
254     return 0;
255 }
256
257
258 ##### EX 13 #####
259
260
261     /*****/
262     /* Table of */
263     /* Sine Function */
264     /*****/
265
266     /* Michel Vallieres */
267     /* Written: Winter 1995 */
268 #include <stdio.h>
269 #include <math.h>
270
271 int main()
272 {

```

```

273     int     angle_degree;
274     double  angle_radian, pi, value;
275
276             /* Print a header */
277     printf ("\nCompute a table of the sine function\n\n");
278
279             /* obtain pi once for all */
280             /* or just use pi = M_PI, where
281 *             M_PI is defined in math.h */
282     pi = 4.0*atan(1.0);
283     printf ( " Value of PI = %f \n\n", pi );
284
285     printf ( " angle      Sine \n" );
286
287     angle_degree=0;             /* initial angle value */
288             /* scan over angle */
289
290     while ( angle_degree <= 360 ) /* loop until angle_degree > 360 */
291     {
292         angle_radian = pi * angle_degree/180.0 ;
293         value = sin(angle_radian);
294         printf ( " %3d      %f \n ", angle_degree, value );
295
296         angle_degree = angle_degree + 10; /* increment the loop index */
297     }
298     return 0;
299 }
300
301 ##### EX 14 #####
302
303 /* the program converts the given weight from kilograms to pounds */
304
305 #include<stdio.h>
306
307 int main(){
308
309     float kg_val, pounds_val;
310     printf("Enter a weight in kg: \n"); // display a message for user
311     scanf("%f",&kg_val); // read the input value as real numbers
312     precision
313     pounds_val=kg_val*2.2; // compute the value in pounds
314     printf("\n\n%6.2f kilos = %6.2f pounds \n", kg_val, pounds_val); // display the
315     input and output values
316     return 0; // return from main
317 }
318

```