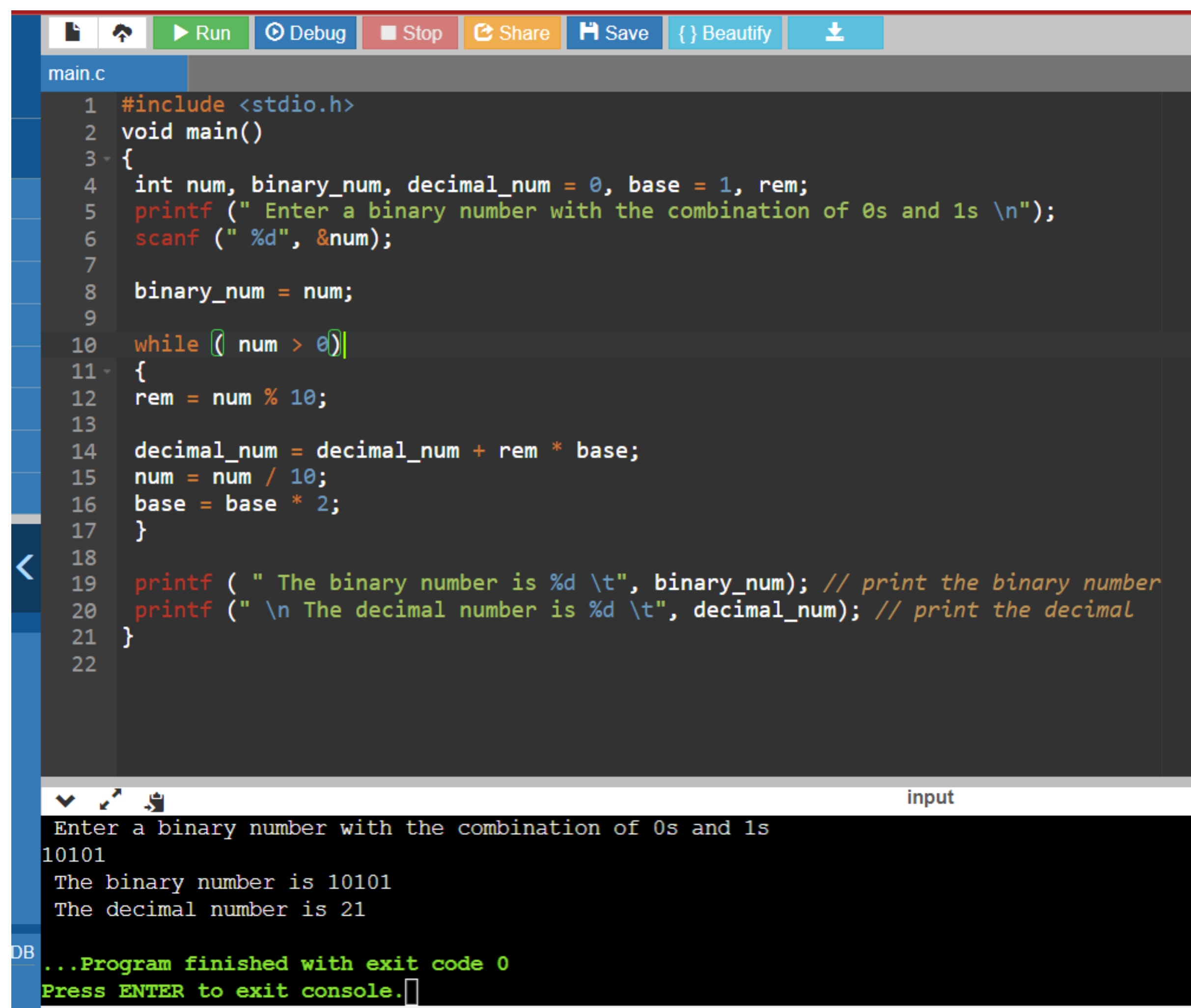


## Binary to decimal conversions



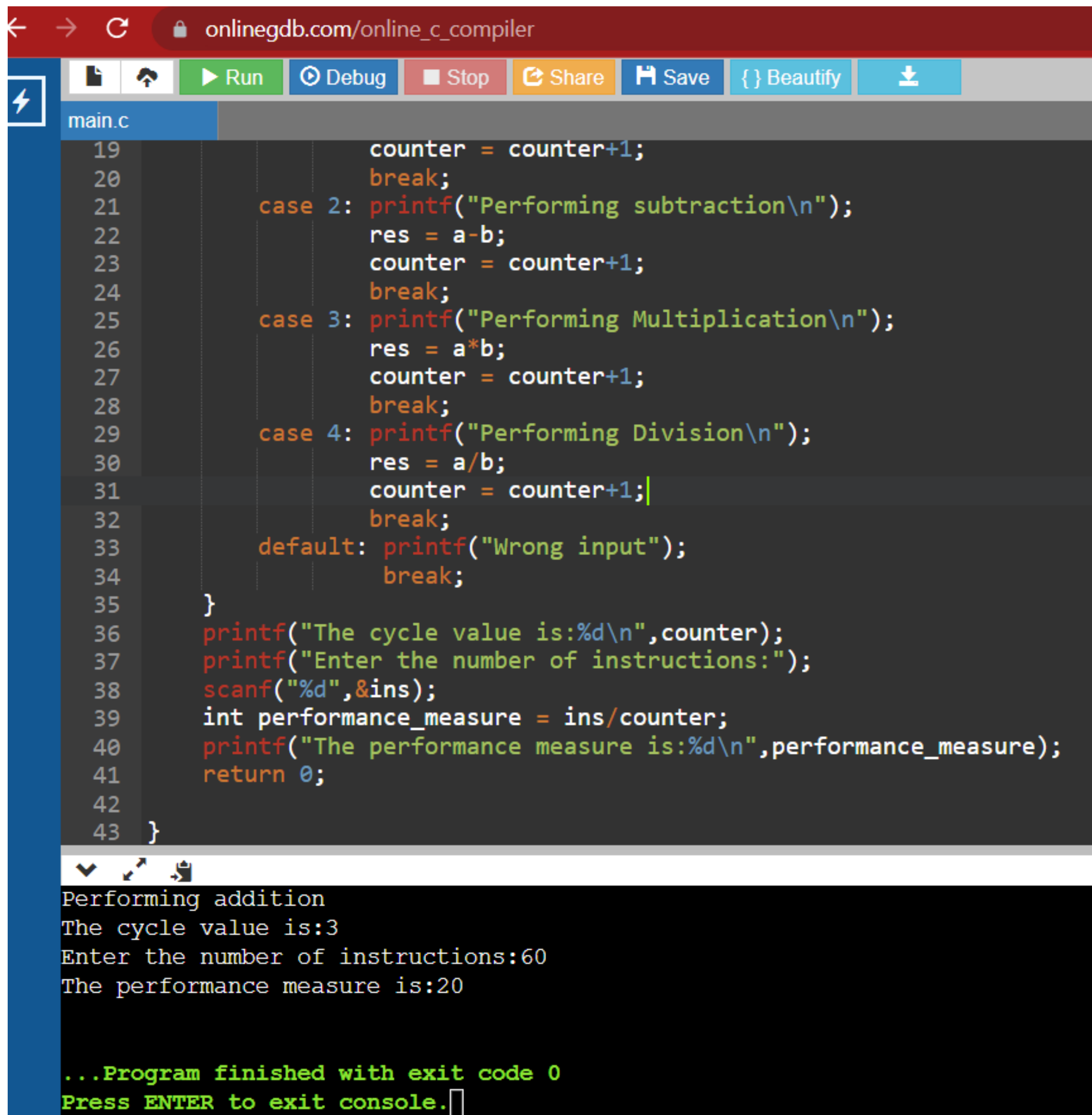
The image shows a screenshot of a code editor with a C program for converting binary to decimal. The editor has a toolbar at the top with buttons for Run, Debug, Stop, Share, Save, Beautify, and a download icon. The file name is 'main.c'. The code is as follows:

```
1 #include <stdio.h>
2 void main()
3 {
4     int num, binary_num, decimal_num = 0, base = 1, rem;
5     printf (" Enter a binary number with the combination of 0s and 1s \n");
6     scanf ("%d", &num);
7
8     binary_num = num;
9
10    while (( num > 0))
11    {
12        rem = num % 10;
13
14        decimal_num = decimal_num + rem * base;
15        num = num / 10;
16        base = base * 2;
17    }
18
19    printf ( " The binary number is %d \t", binary_num); // print the binary number
20    printf (" \n The decimal number is %d \t", decimal_num); // print the decimal
21 }
22
```

Below the code editor is a console window. The input '10101' is entered. The output shows the binary number '10101' and the decimal number '21'. The program finishes with exit code 0, and the console prompts the user to press ENTER to exit.

```
input
Enter a binary number with the combination of 0s and 1s
10101
The binary number is 10101
The decimal number is 21
...Program finished with exit code 0
Press ENTER to exit console.
```

## 2 stage pipe lining



The screenshot shows a web browser window with the URL `onlinegdb.com/online_c_compiler`. The interface includes a toolbar with buttons for Run, Debug, Stop, Share, Save, Beautify, and a download icon. A file named `main.c` is open, displaying the following C code:

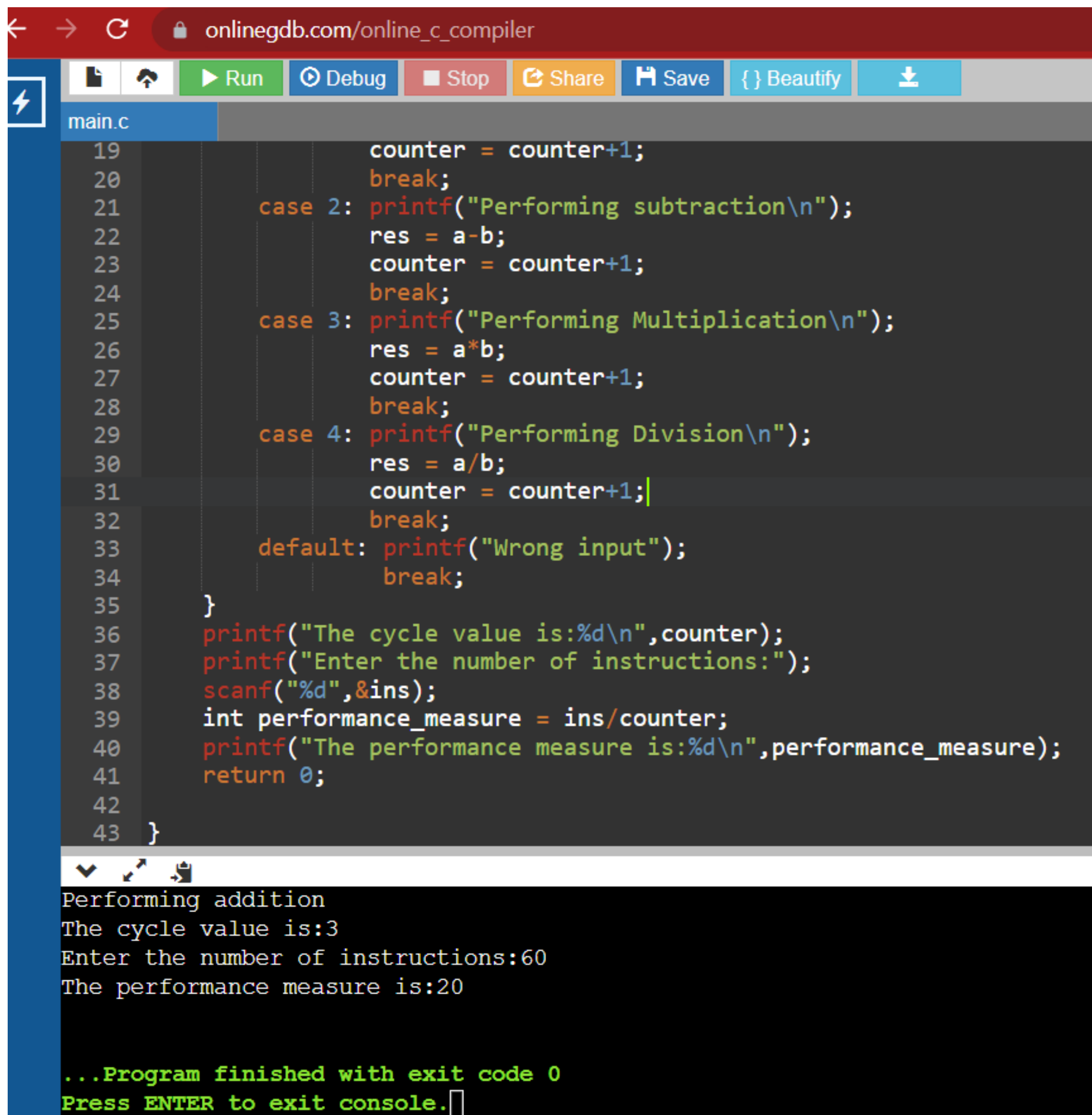
```
19         counter = counter+1;
20         break;
21     case 2: printf("Performing subtraction\n");
22             res = a-b;
23             counter = counter+1;
24             break;
25     case 3: printf("Performing Multiplication\n");
26             res = a*b;
27             counter = counter+1;
28             break;
29     case 4: printf("Performing Division\n");
30             res = a/b;
31             counter = counter+1;
32             break;
33     default: printf("Wrong input");
34             break;
35 }
36 printf("The cycle value is:%d\n",counter);
37 printf("Enter the number of instructions:");
38 scanf("%d",&ins);
39 int performance_measure = ins/counter;
40 printf("The performance measure is:%d\n",performance_measure);
41 return 0;
42
43 }
```

Below the code editor, the execution output is shown in a console window:

```
Performing addition
The cycle value is:3
Enter the number of instructions:60
The performance measure is:20

...Program finished with exit code 0
Press ENTER to exit console.
```

3 stage pipe lining



The screenshot shows a web browser window with the URL `onlinegdb.com/online_c_compiler`. The interface includes a toolbar with buttons for Run, Debug, Stop, Share, Save, Beautify, and a download icon. A file named `main.c` is open, displaying the following C code:

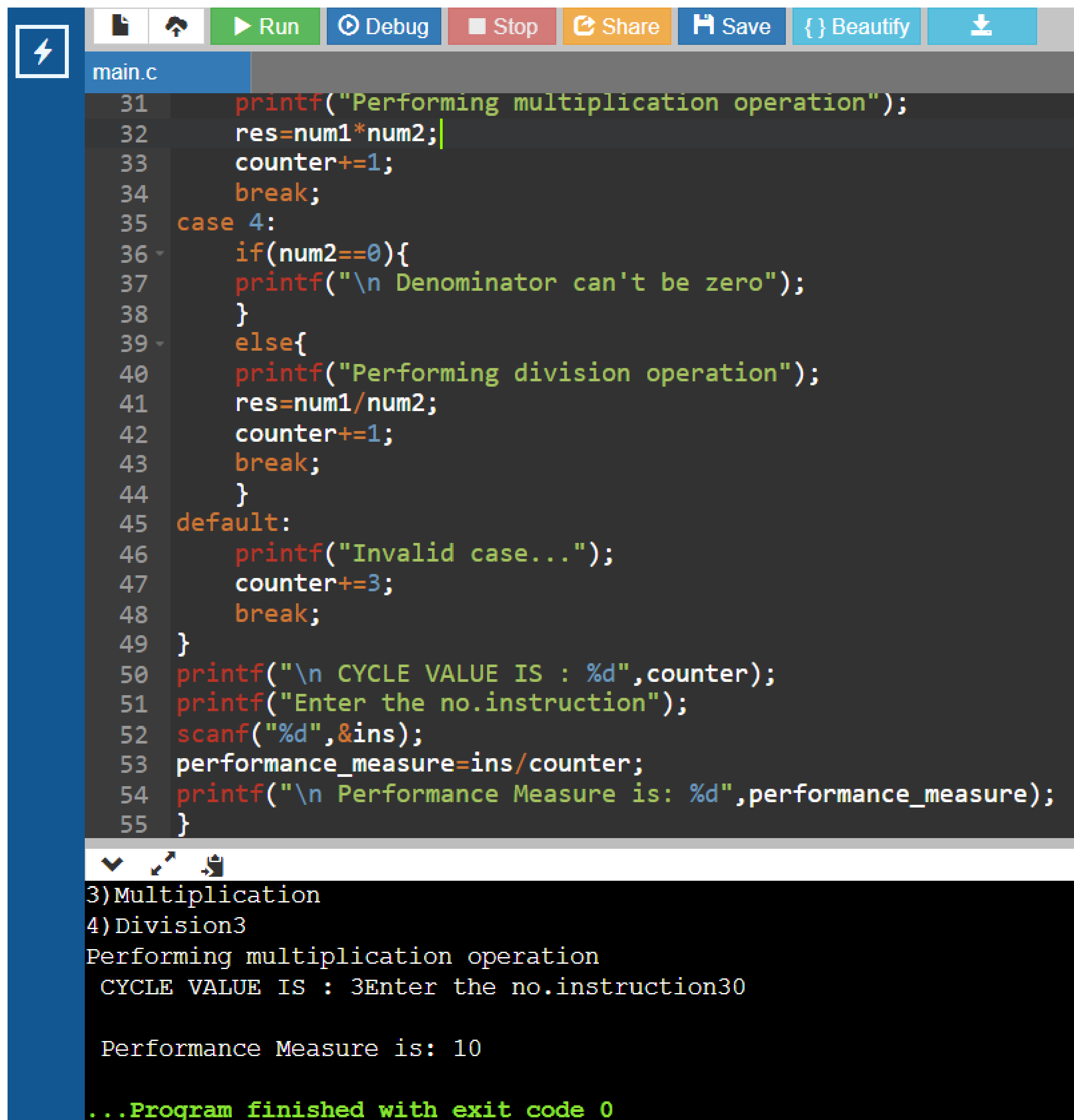
```
19     counter = counter+1;
20     break;
21     case 2: printf("Performing subtraction\n");
22             res = a-b;
23             counter = counter+1;
24             break;
25     case 3: printf("Performing Multiplication\n");
26             res = a*b;
27             counter = counter+1;
28             break;
29     case 4: printf("Performing Division\n");
30             res = a/b;
31             counter = counter+1;
32             break;
33     default: printf("Wrong input");
34             break;
35 }
36 printf("The cycle value is:%d\n",counter);
37 printf("Enter the number of instructions:");
38 scanf("%d",&ins);
39 int performance_measure = ins/counter;
40 printf("The performance measure is:%d\n",performance_measure);
41 return 0;
42
43 }
```

Below the code editor, the execution output is shown in a console window:

```
Performing addition
The cycle value is:3
Enter the number of instructions:60
The performance measure is:20

...Program finished with exit code 0
Press ENTER to exit console.
```

4 stage pipelining



```
main.c
31 printf("Performing multiplication operation");
32 res=num1*num2;
33 counter+=1;
34 break;
35 case 4:
36 if(num2==0){
37 printf("\n Denominator can't be zero");
38 }
39 else{
40 printf("Performing division operation");
41 res=num1/num2;
42 counter+=1;
43 break;
44 }
45 default:
46 printf("Invalid case...");
47 counter+=3;
48 break;
49 }
50 printf("\n CYCLE VALUE IS : %d",counter);
51 printf("Enter the no.instruction");
52 scanf("%d",&ins);
53 performance_measure=ins/counter;
54 printf("\n Performance Measure is: %d",performance_measure);
55 }
```

3)Multiplication  
4)Division3  
Performing multiplication operation  
CYCLE VALUE IS : 3Enter the no.instruction30  
  
Performance Measure is: 10  
...Program finished with exit code 0

Booth algorithm

main.c

```
138 printf("\n-->");
139 arshift();
140 q = anum[i];
141 }
142 else if(anum[i] == 1 && q == 0){//subtract and shift for 10
143 printf("\n-->");
144 printf("\nSUB B: ");
145 add(bcomp);//add two's complement to implement subtraction
146 arshift();
147 q = anum[i];
148 }
149 else{//add ans shift for 01
150 printf("\n-->");
151 printf("\nADD B: ");
152 add(bnum);
153 arshift();
154 q = anum[i];
155 }
156 }
157 printf("\nProduct is = ");
158 for (i = 4; i >= 0; i--){
159 printf("%d", pro[i]);
160 }
161 for (i = 4; i >= 0; i--){
162 printf("%d", anumcp[i]);
```

SUB B: 11010:00011  
AR-SHIFT: 11101:00001  
-->  
AR-SHIFT: 11110:10000  
Product is = 1111010000

...Program finished with exit code 1  
Press ENTER to exit console.

Restoration

main.c

```
1  #include<stdlib.h>
2  #include<stdio.h>
3  int acum[100]={0} ;
4  void add(int acum[],int b[],int n);
5  int q[100],b[100];
6  int main()
7  {
8  int x,y;
9  printf("Enter the Number :");
10 scanf("%d%d",&x,&y);
11 int i=0;
12 while(x>0 || y>0)
13 {
14 if(x>0)
15 {
16 q[i]=x%2;
17 x=x/2;
18 }
19 else
20 {
21 q[i]=0;
22 }
23 if(y>0)
24 {
25 b[i]=y%2;
```

Enter the Number :30  
11

Quoient : 00010  
Remainder : 001000

...Program finished with exit code 0

CPU PERFORMANCE

Run

Debug

Stop

Share

Save

Beautify

main.c

```
1  #include <stdio.h>
2  int main()
3  {
4      float cr;
5      int p,p1,i;
6      float cpu[5];
7      float cpi,ct,max;
8      int n=1000;
9      for(i=0;i<=4;i++)
10     {
11         cpu[i]=0;
12     }
13     printf("\n Enter the number of processors:");
14     scanf("%d",&p);
15     p1=p;
16     for(i=0;i<p;i++)
17     {
18         printf("\n Enter the Cycles per Instrcution of processor:");
19         scanf("%f",&cpi);
20         printf("\n Enter the clockrate in GHz:");
21         scanf("%f",&cr);
22         ct=1000*cpi/cr;
23         printf("The CPU time is: %f",ct);
24         cpu[i]=ct;
25     }
```

Enter the Cycles per Instrcution of processor:34  
  
Enter the clockrate in GHz:3425  
The CPU time is: 9.927008  
The processor has lowest Execution time is: 3.871967  
  
...Program finished with exit code 0  
Press ENTER to exit console.