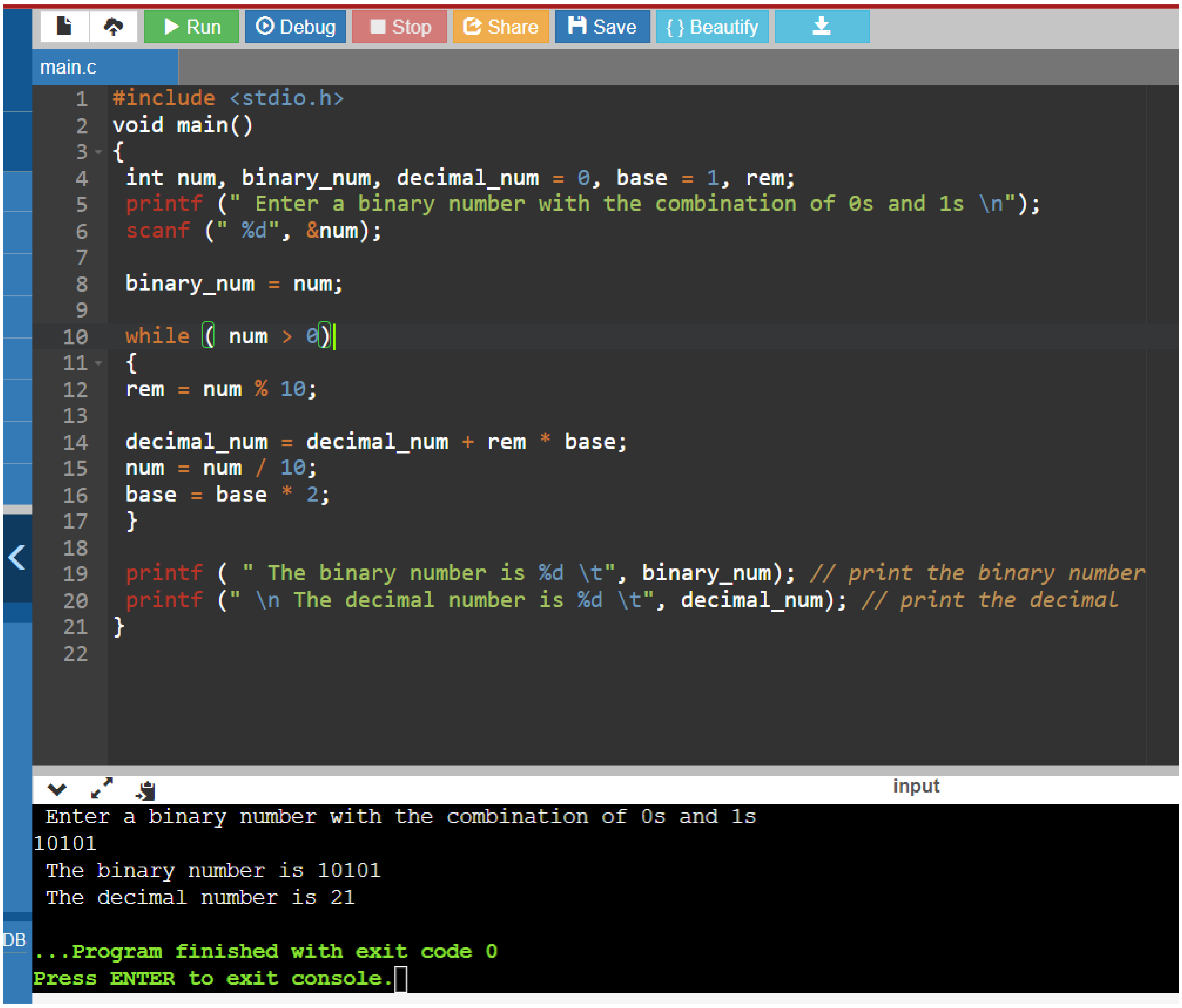
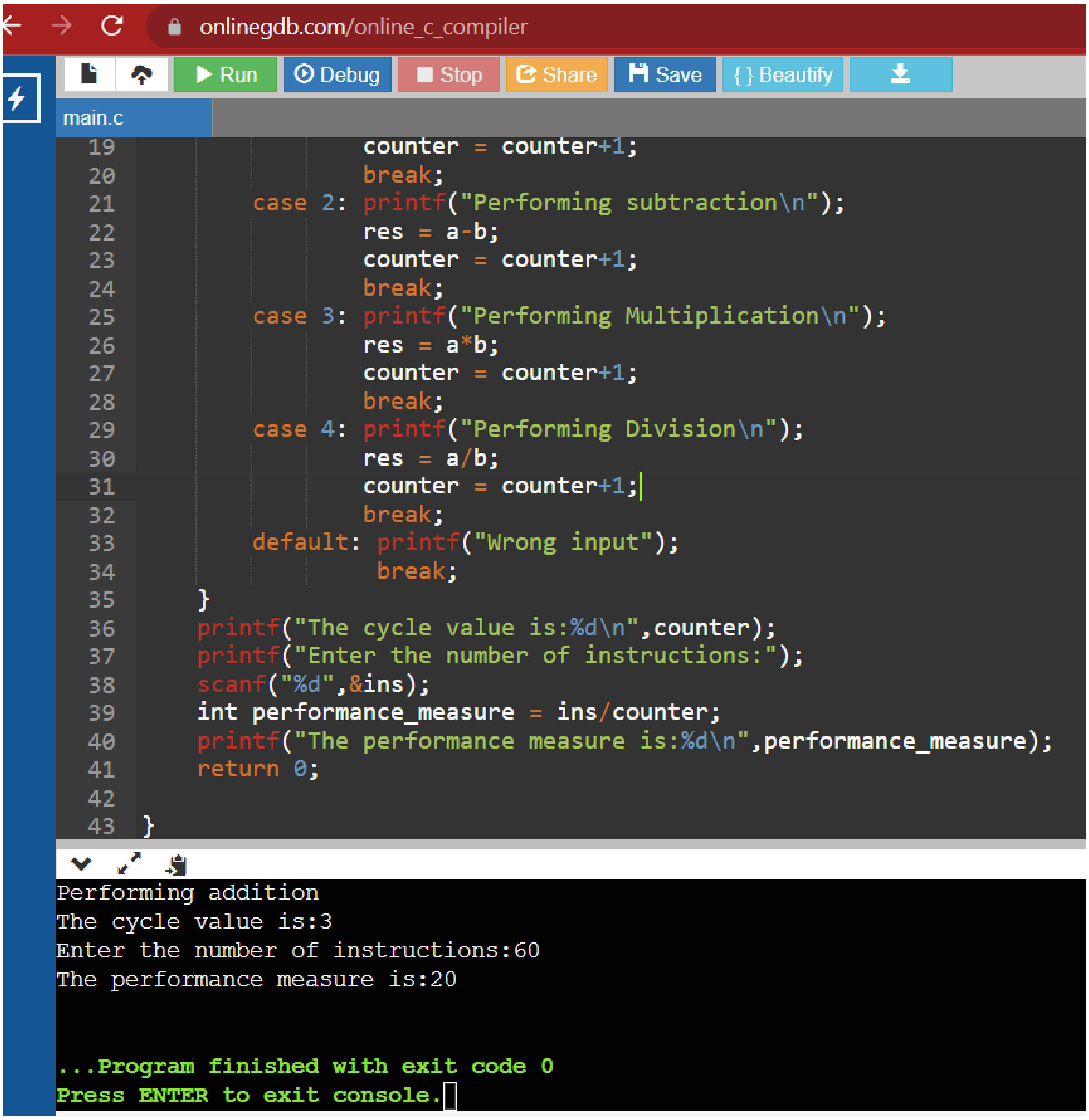
Binary to decimal conversions



```c
#include <stdio.h>
void main()
{
 int num, binary_num, decimal_num = 0, base = 1, rem;
 printf (" Enter a binary number with the combination of 0s and 1s \n");
 scanf (" %d", &num);

 binary_num = num;

 while ( num > 0)
 {
 rem = num % 10;

 decimal_num = decimal_num + rem * base;
 num = num / 10;
 base = base * 2;
 }

 printf ( " The binary number is %d \t", binary_num); // print the binary number
 printf (" \n The decimal number is %d \t", decimal_num); // print the decimal
}

```

```
 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.
```

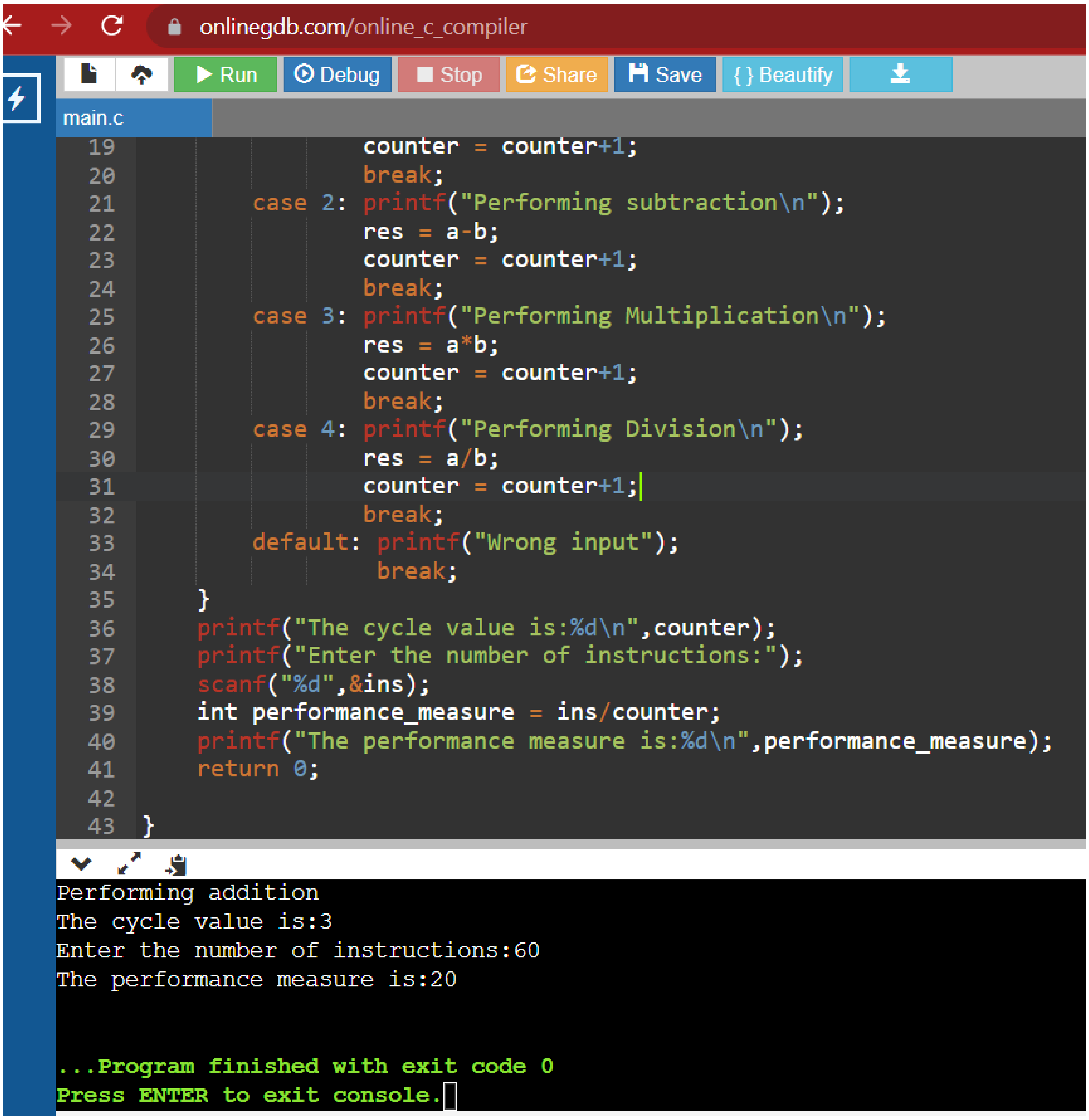
2 stage pipe lining



```c
                counter = counter+1;
                break;
        case 2: printf("Performing subtraction\n");
                res = a-b;
                counter = counter+1;
                break;
        case 3: printf("Performing Multiplication\n");
                res = a*b;
                counter = counter+1;
                break;
        case 4: printf("Performing Division\n");
                res = a/b;
                counter = counter+1;
                break;
        default: printf("Wrong input");
                 break;
    }
    printf("The cycle value is:%d\n",counter);
    printf("Enter the number of instructions:");
    scanf("%d",&ins);
    int performance_measure = ins/counter;
    printf("The performance measure is:%d\n",performance_measure);
    return 0;

}
```

```
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.
```

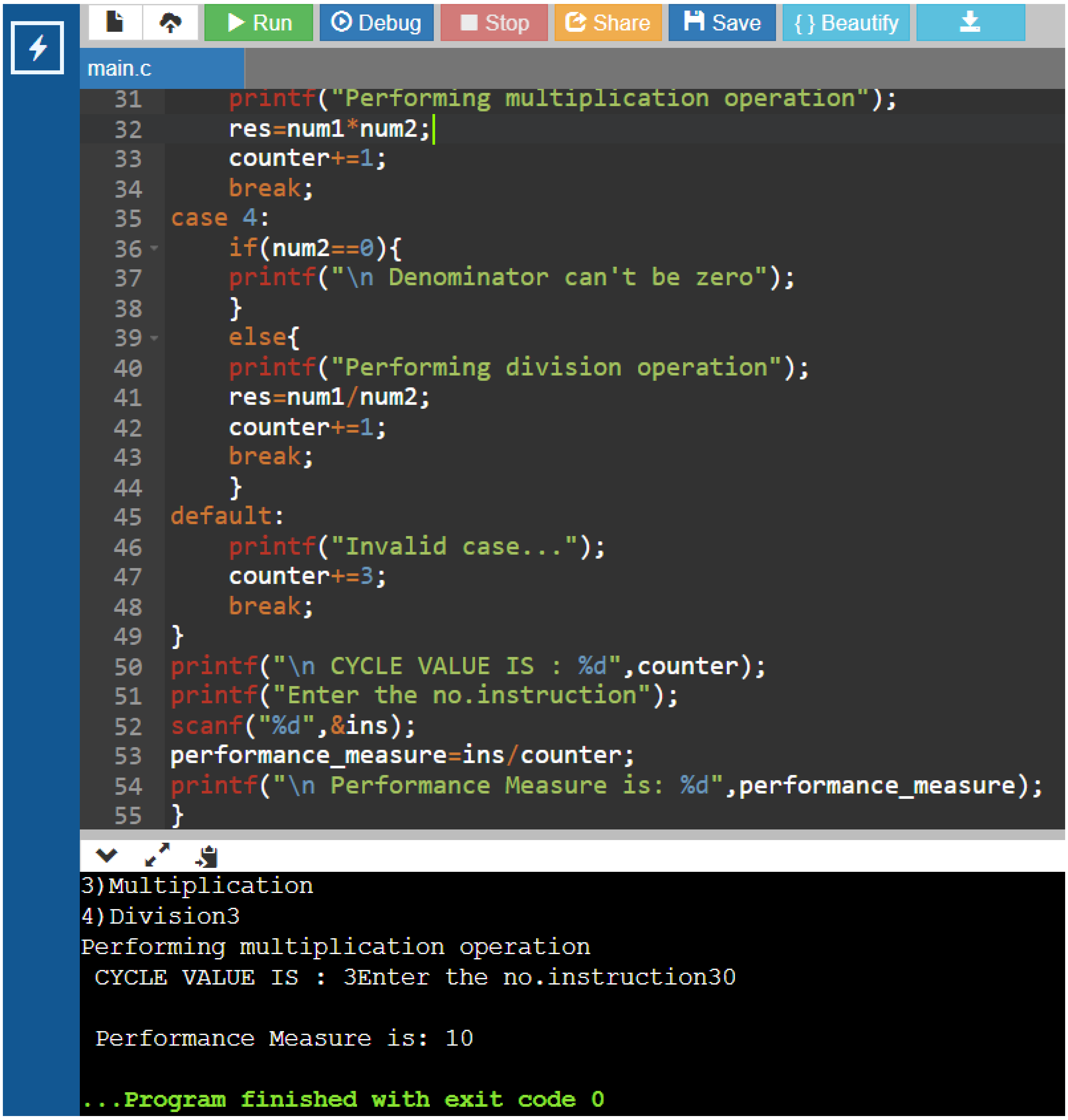
3 stage pipe lining



```c
                counter = counter+1;
                break;
        case 2: printf("Performing subtraction\n");
                res = a-b;
                counter = counter+1;
                break;
        case 3: printf("Performing Multiplication\n");
                res = a*b;
                counter = counter+1;
                break;
        case 4: printf("Performing Division\n");
                res = a/b;
                counter = counter+1;
                break;
        default: printf("Wrong input");
                 break;
    }
    printf("The cycle value is:%d\n",counter);
    printf("Enter the number of instructions:");
    scanf("%d",&ins);
    int performance_measure = ins/counter;
    printf("The performance measure is:%d\n",performance_measure);
    return 0;

}
```

```
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.
```

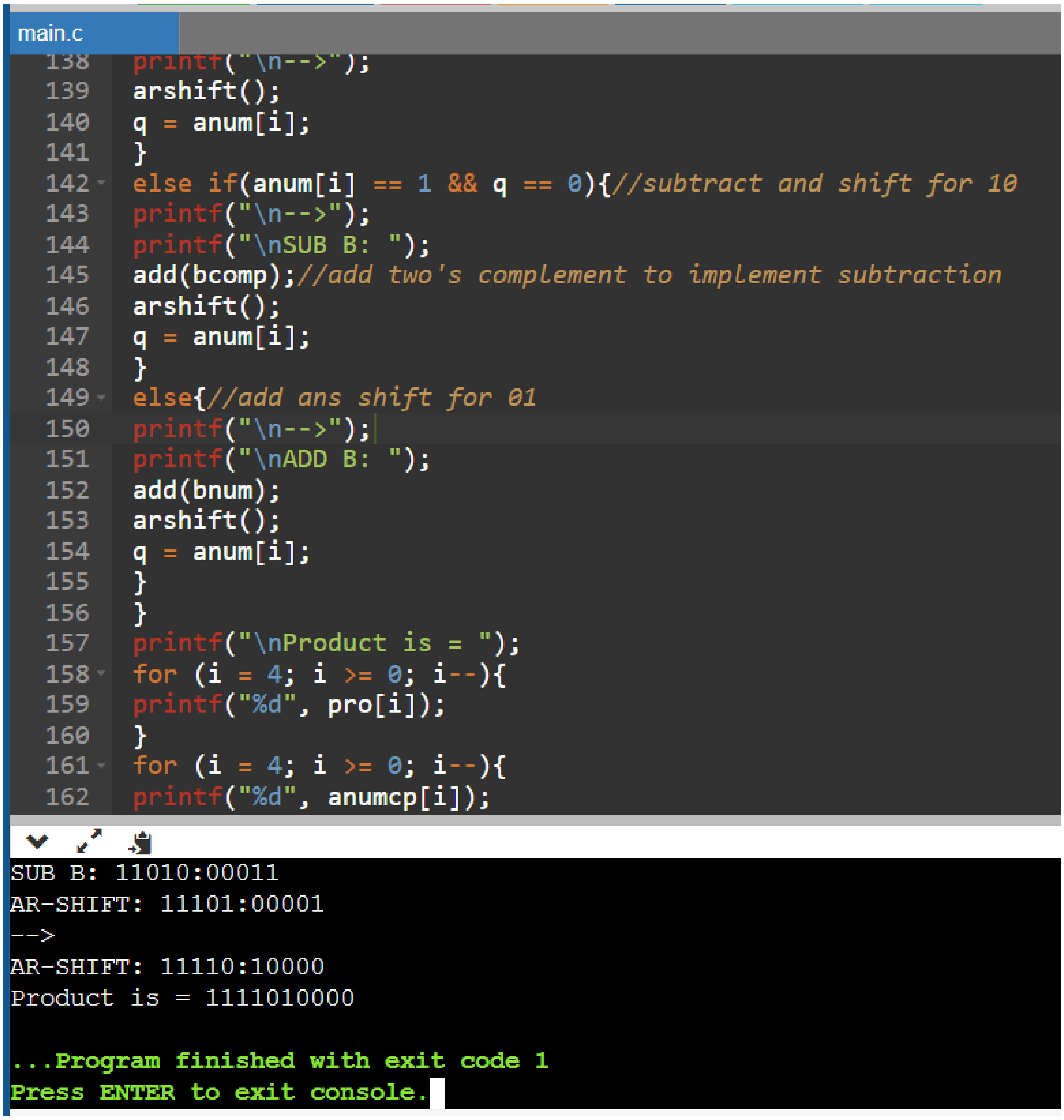
4 stage pipelining



```c
    printf("Performing multiplication operation");
    res=num1*num2;
    counter+=1;
    break;
case 4:
    if(num2==0){
    printf("\n Denominator can't be zero");
    }
    else{
    printf("Performing division operation");
    res=num1/num2;
    counter+=1;
    break;
    }
default:
    printf("Invalid case...");
    counter+=3;
    break;
}
printf("\n CYCLE VALUE IS : %d",counter);
printf("Enter the no.instruction");
scanf("%d",&ins);
performance_measure=ins/counter;
printf("\n Performance Measure is: %d",performance_measure);
}
```

```
3)Multiplication
4)Division3
Performing multiplication operation
 CYCLE VALUE IS : 3Enter the no.instruction30

 Performance Measure is: 10

...Program finished with exit code 0
```

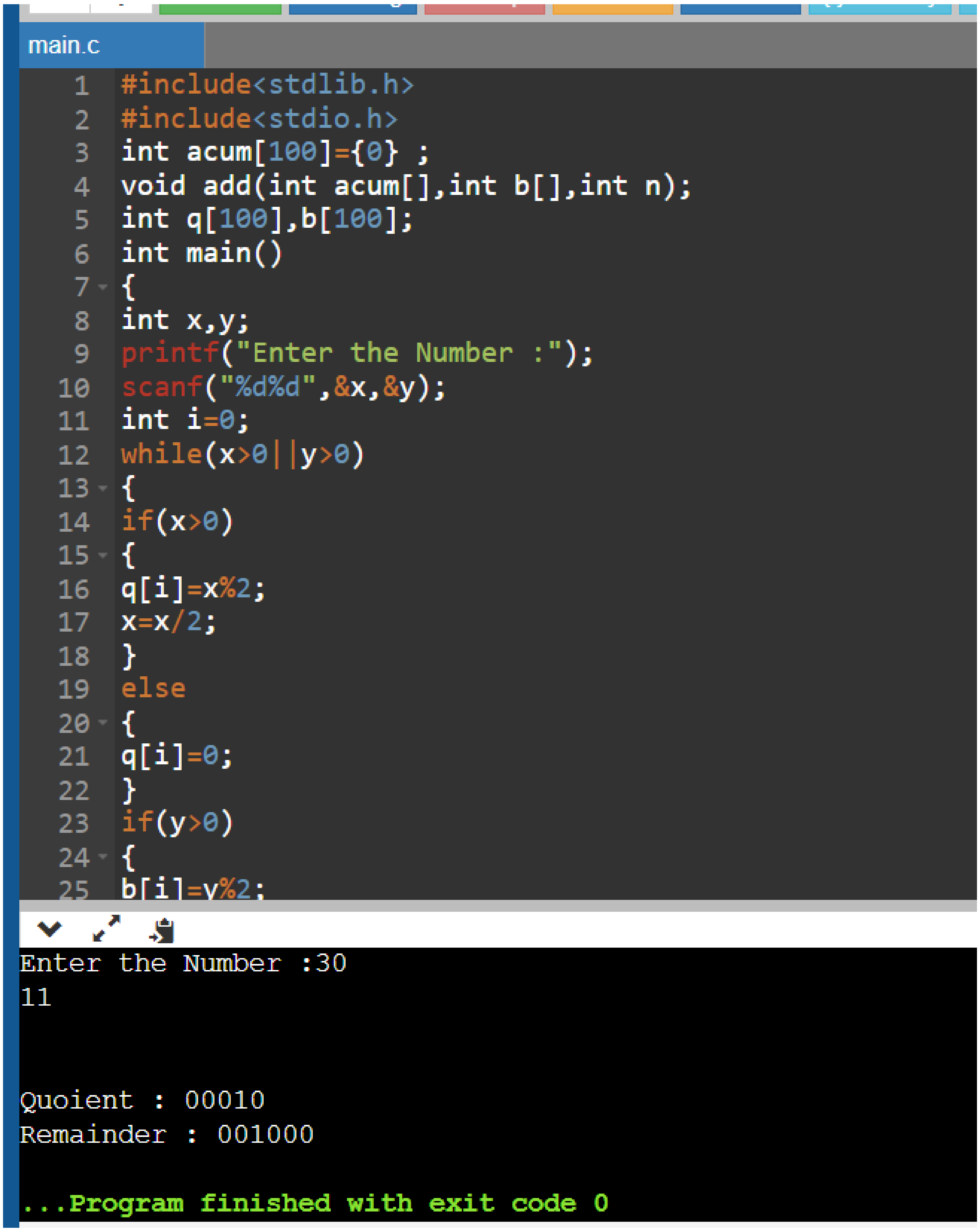
Booth algorithem



```c
printf("\n-->");
arshift();
q = anum[i];
}
else if(anum[i] == 1 && q == 0){//subtract and shift for 10
printf("\n-->");
printf("\nSUB B: ");
add(bcomp);//add two's complement to implement subtraction
arshift();
q = anum[i];
}
else{//add ans shift for 01
printf("\n-->");
printf("\nADD B: ");
add(bnum);
arshift();
q = anum[i];
}
}
printf("\nProduct is = ");
for (i = 4; i >= 0; i--){
printf("%d", pro[i]);
}
for (i = 4; i >= 0; i--){
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.
```

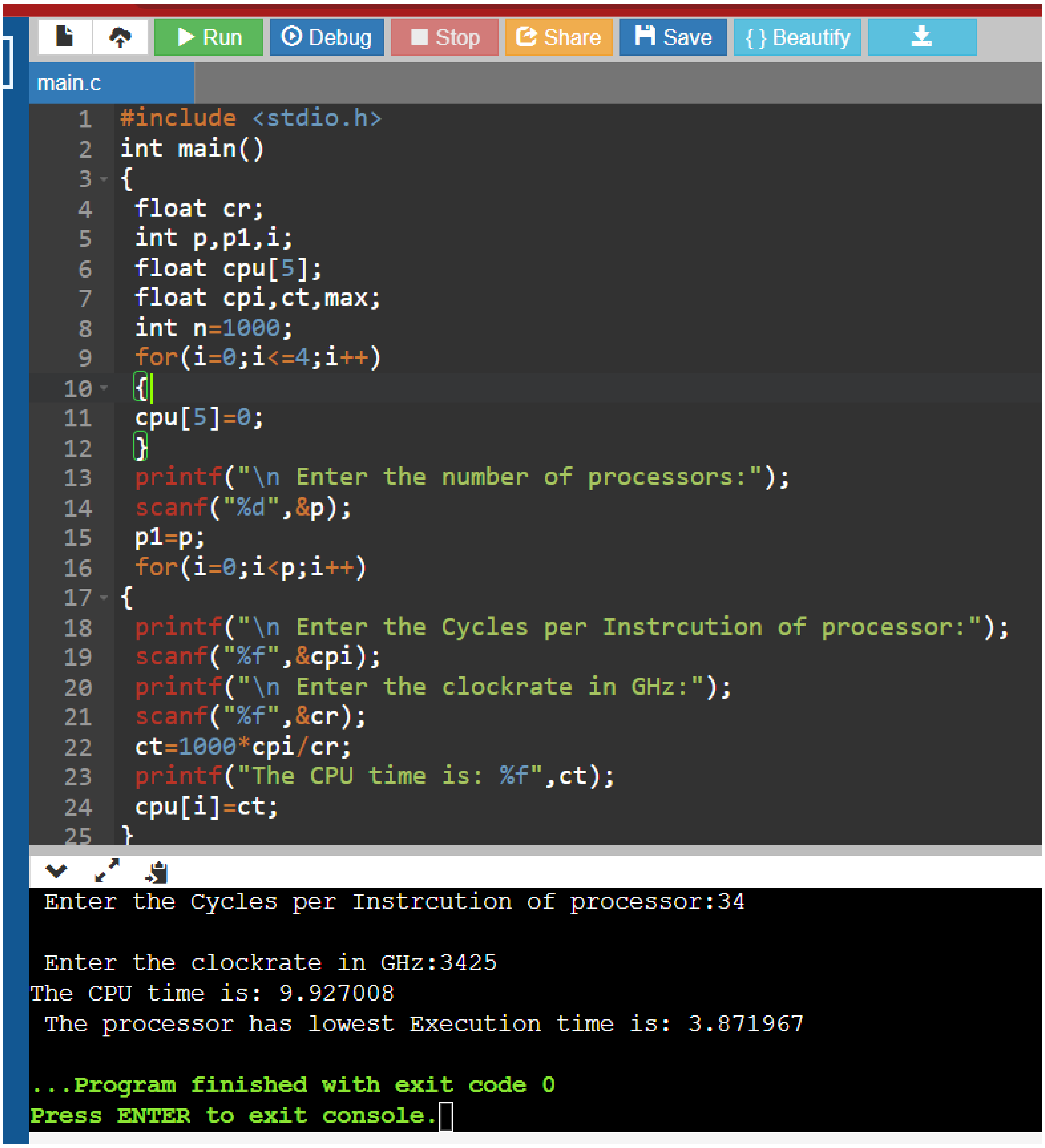
Restoration



```c
#include<stdlib.h>
#include<stdio.h>
int acum[100]={0} ;
void add(int acum[],int b[],int n);
int q[100],b[100];
int main()
{
int x,y;
printf("Enter the Number :");
scanf("%d%d",&x,&y);
int i=0;
while(x>0||y>0)
{
if(x>0)
{
q[i]=x%2;
x=x/2;
}
else
{
q[i]=0;
}
if(y>0)
{
b[i]=y%2;
```

```
Enter the Number :30
11


Quoient : 00010
Remainder : 001000

...Program finished with exit code 0
```

CPU PERFORMANCE



```c
#include <stdio.h>
int main()
{
 float cr;
 int p,p1,i;
 float cpu[5];
 float cpi,ct,max;
 int n=1000;
 for(i=0;i<=4;i++)
 {
 cpu[5]=0;
 }
 printf("\n Enter the number of processors:");
 scanf("%d",&p);
 p1=p;
 for(i=0;i<p;i++)
{
 printf("\n Enter the Cycles per Instrcution of processor:");
 scanf("%f",&cpi);
 printf("\n Enter the clockrate in GHz:");
 scanf("%f",&cr);
 ct=1000*cpi/cr;
 printf("The CPU time is: %f",ct);
 cpu[i]=ct;
}
```

```
 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.
```